教学与多媒体开发平台说明书

1　概述

多媒体教学与开发平台是专为大学电子学科类本科生、研究生和科研院所的教学及科研准备的FPGA 开发平台。

平台可以作为：

1. 电子、通信和计算机类专业本科生及研究生数字电路、多媒体、通信及 IC 设计等课程的教学平台；

2. 计算机、微电子、通信、多媒体、测控、自动化及 IC 设计领域科研院所及相关企业的研究设计及验证平台。

1.1　核心板

核心板是由 FPGA 及一些常用的外围器件组成的一个最小的也是完整的 SoC 系统（图 1），包括FPGA、存储器、网络接口及串口等。核心板的设计首先提供一个完整的初中级数字电路及 SoC 的教学及设计平台，同时提供一种模块化的设计思路。核心板上的所有器件都是常用的器件，为大部分的用户应用所必需，同时具备低成本的特点，可以作为用户设计的一个低成本模块加入到用户应用中，简化应用设计。

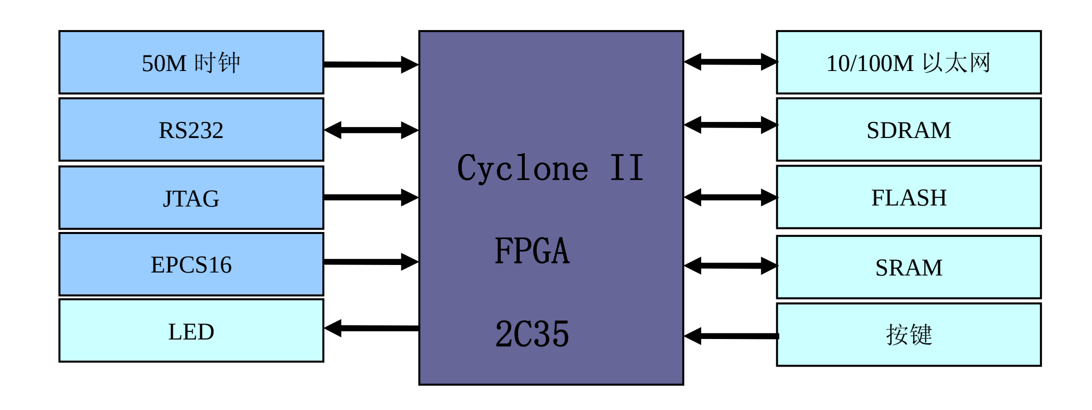
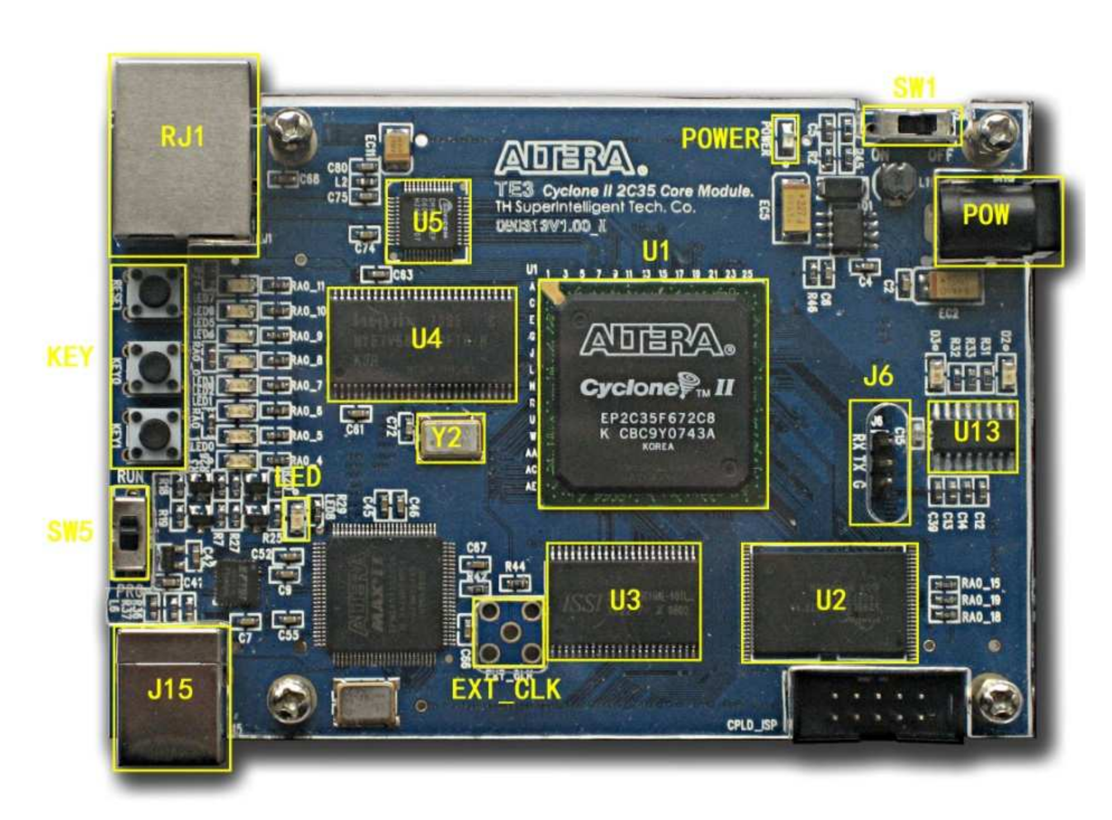
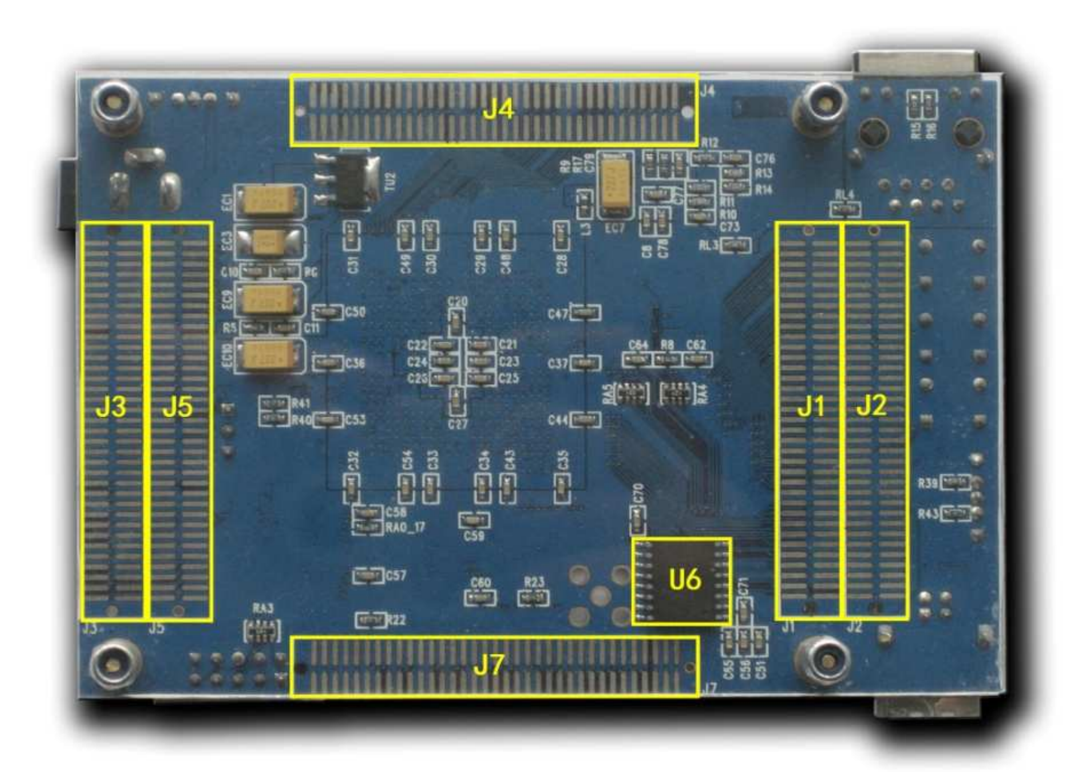


图 1　核心板框图



(a) 核心版正面



(b) 核心版反面

图 2 核心板照片

1.1.1 主要特性

1. 具有网络接口的最小且完整 SOPC 系统；
2. 丰富的存储器资源；
3. 通过 6 个高速接口为用户提供 380 个 I/O 引脚；
4. 体积小（97mm*74mm）,易携带；
5. SMA 时钟接口；
6. 支持 JTAG 和 AS 配置；
7. 上下玻璃罩保证美观及安全性。

1.1.2 核心板的主要器件(图 2)

1. Cyclone II 2C35 FPGA（图 2 U1）；
2. EPCS16 配置芯片（图 2 U6）；
3. JTAG 及 AS 下载接口（图 2 J15）；
4. 32M 字节 SDRAM（图 2 U4）；
5. 8M 字节 Flash（图 2 U2）；
6. 512K 字节 SRAM（图 2 U3）；
7. 10/100M 以太网控制器 DM9000A（图 2 U5）及网卡接口（图 2 RJ1）；
8. 3 个按键 KEY0～KEY1、RESET（图 2 KEY）；
9. 8 个黄绿色 LED 灯 LED0～LED7（图 2 LED）；
10. RS232 接口（图 2 J6）；
11. 50MHz 晶振（图 2 Y2）及 SMA 时钟接口（图 2 EXT_CLK）；
12. 6 个高速 70pin 扩展接口(图 2 J1/J2/J3/J4/J5/J7);
13. 提供高可靠、高稳定的板级及芯片级电压,可以在核心板单独使用时提供供电，也可以从扩展板供电。

1.2 应用领域

1. 初中级数字及系统设计
2. SOPC 设计
3. 作为用户应用的核心模块，降低用户设计复杂性

1.3 扩展板

由于核心板采用模块化的接口，用户可以根据自身应用设计扩展板，目前开发的扩展板是配合当前各高校教学改革及宽带无线多媒体的开发而设计的。扩展板具有丰富的用户

接口，除了常用的输入输出接口之外，为模拟音视频设计了 AV、VGA 及音频输入输出接口，为数字视频及无线应用增加了 ASI、A/D 及 D/A 接口。

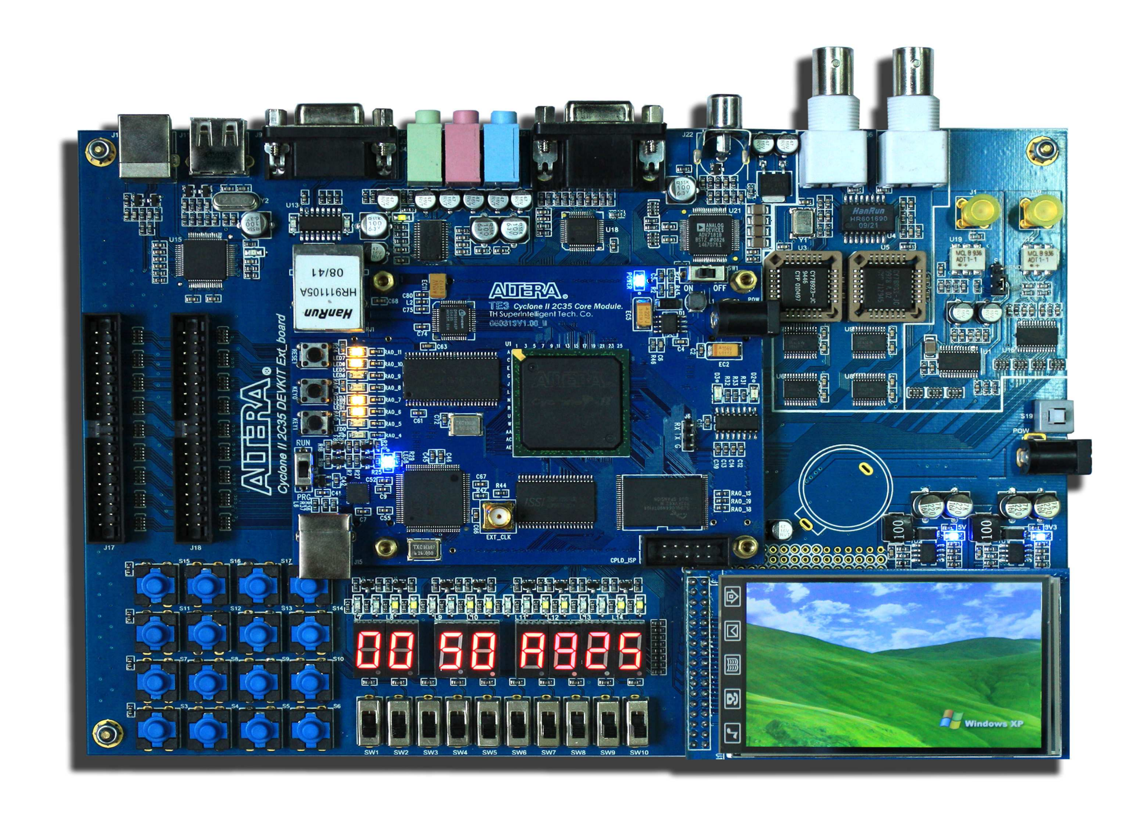


图 3　扩展板和核心板组成的教学开发平台

1.3.1 主要特性

1. 完整的数字设计、SOPC、多媒体及无线通信平台；
2. 丰富的音视频接口(CVBS、VGA、Audio In/Out 及 Mic In)；
3. 无线通信设计所需的 A/D 及 D/A 接口；
4. 数字电视和 IPTV 设计所需的 TS 和以太网接口；
5. 丰富的用户输入、输出接口（按键、开关、数码管、LED 和 LCD）。

1.3.2 主要器件及接口

1. 10 个拨动开关和 16 个按键；
2. 8 个七段数码管和 16 个 LED 灯；
3. USB 接口（Host 和 Slave）；
4. RS232 接口；
5. 24 位 CD 品质音频的编/解码器 UDA1341TS，带有麦克风的输入插座、音频输入插座和音频输出插座。

6. VGA DAC ADV7132（内含三个 10 位高速 DAC）及 VGA 输出接口；
7. 视频 AV 输入接口；
8. 10bit 40MHz A/D；
9. 14bit 165MHz D/A；
10. 128*64 像素 LCD 模块；
11. ASI 收发器及输入输出接口；
12. 802.11 无线模块。

1.3.3 应用领域

1. 高级数字及接口设计
2. 复杂 SOPC 设计
3. 多媒体音视频处理
4. 无线多媒体应用（DTV, IPTV, WiMAX 等）

2 计算机软硬件要求

2.1 硬件要求

1. 主频：1.6G 以上；
2. 内存：至少 512M 字节，推荐大于 1G 字节；
3. 硬盘：大于 4G 的剩余空间；
4. 接口：USB2.0 接口 1 个。

2.2 软件要求

1. 操作系统：Windows XP SP2 以上；
2. 编译综合工具：Altera Quartus II 7.0；
3. 综合工具：MentorGraphics ModelSim6.2e；
4. uClinux: Microtronix uClinux 1.4.1（附件中包含）。

注意：本产品中所有程序都通过以上版本的软硬件验证，如在其他软硬件版本上使用可能需要做一些修改。

3　硬件连接及使用

3.1　核心版连接及使用方法

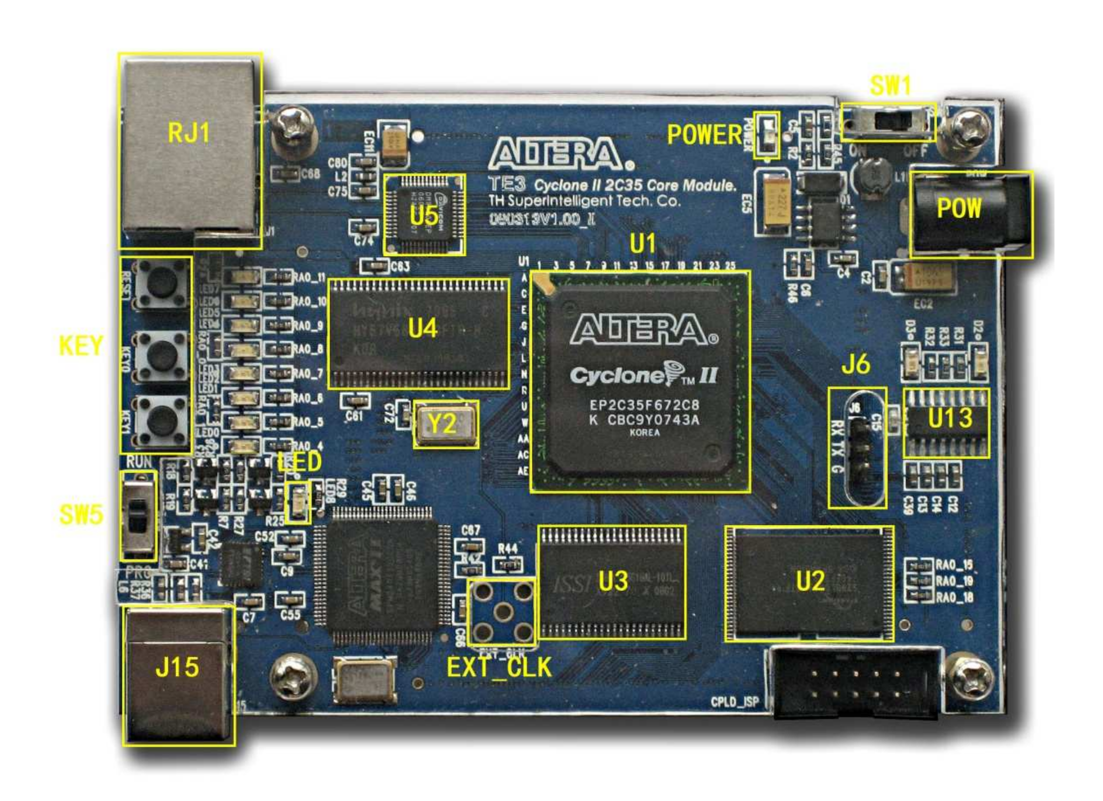


图 4　核心版正面照片

1. 在使用开发平台之前首先安装 Quartus II 7.0。
2. 确保电源开关（图 4 SW1）在 OFF 状态，即在右端。
3. 将电源插入右上角电源接口上（图 4 POW）。
4. 将 USB 线方头插入 USB 接口（图 4 J15）中，另一头与计算机的 USB 接口相连；
5. 如需使用串口，将串口线小的一头连至串口接口（图 4 J6），另一端连至计算机。
6. 如需使用以太网口，则将以太网双绞线插入以太网接口（图 4 RJ1）中。
7. 连接完毕后，将电源开关（图 4 SW1）拨至 ON 端（即向左拨），电源指示灯（图 4 Power）点亮，核心版开始工作。
8. 如果是第一次使用，电源接通之后，电脑会弹出对话框要求安装下载驱动程序，按下列步骤操作：
   a) 在弹出的新硬件向导（图 5）中选择“否，暂时不”，点击“下一步”；



图 5　找到硬件向导

b) 在新对话框（图 6）中选择“从列表或指定位置安装（高级)”，点击“下一步”；



图 6 找到硬件向导

c) 在新对话框(图 7)中选择“在这些位置上搜索最佳驱动程序”，“在搜索中包括这个位置”，点击浏览；

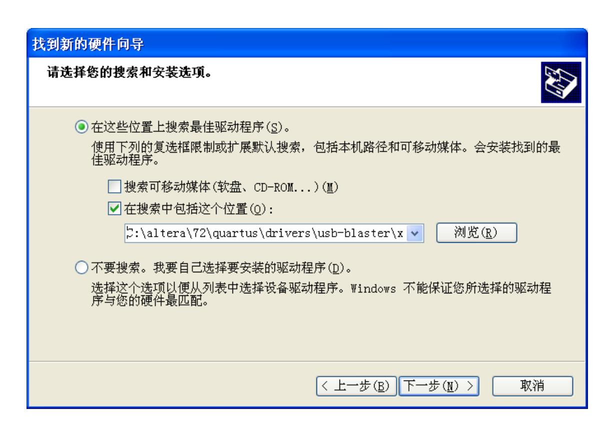


图 7　找到硬件向导

d) 在弹出的浏览对话框（图 8）中选择驱动程序的位置，假设 Quartus 装在了 C 盘根目录下，则驱动程序的位置在：C:\altera\72\quartus\drivers\usb-blaster\x32，然后点击“确定”；

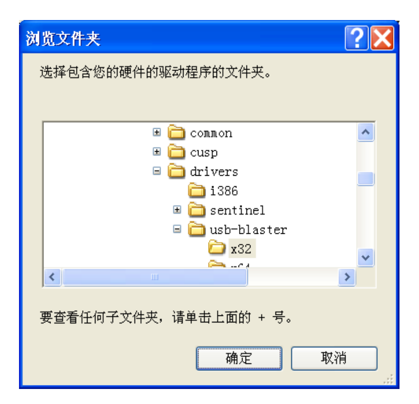


图 8　浏览文件夹

e)　点击确定之后回到下面的对话框（图 9），点击“下一步”；

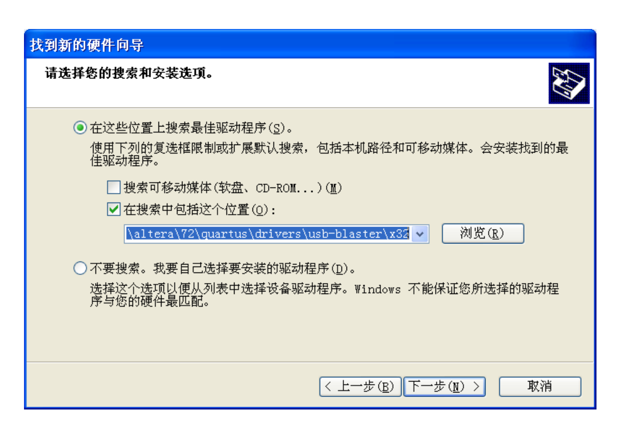


图 9　找到新的硬件向导

f)　然后系统开始寻找并安装驱动程序（图 10），如果弹出下面的对话框（图 11），选“仍然继续”即可；



图 10　找到新的硬件向导

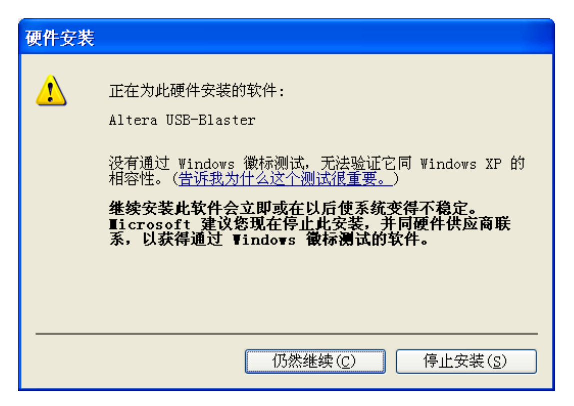


图 11　硬件安装

g)　看出下面所示对话框（图 12）时，表示硬件安装完成，点击“完成”即可。

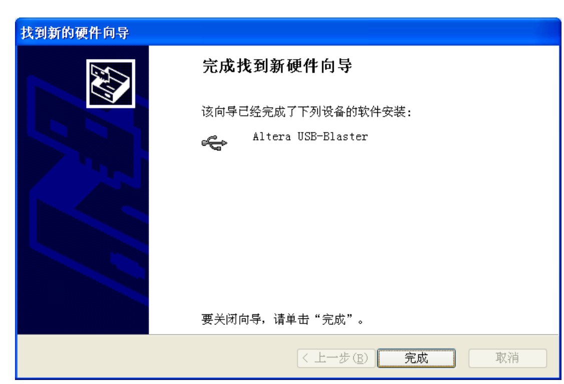


图 12　硬件安装完成

9. 如果使用 Quartus II 对 FPGA 直接编程，将切换开关（图 4 SW5）拨至 RUN 端；如果需要将 FPGA 程序烧制 EPCS 中，则将开关 SW5 拨至 PRG 端；在烧完之后一定要将开关拨回 RUN 端，在下次上电后，EPCS 中的程序会自动下载到 FPGA 中并自动运行。
10. 使用 Quartus II 进行程序设计参见 6.1 节。

3.2 扩展板连接及使用说明

1. 核心板已经在购买之前安装到扩展板上，拆卸核心板请咨询供应商。
2. 开发板上有两个电源插座，分别位于核心板、扩展板上。建议使用扩展板 DC 插座（注：禁止同时使用扩展板和核心板 DC 插座，否则可能造成电路损坏），电源适配器电输出 9V，电源的电流最大 1.5A。扩展板上设有电源有开关。接通电源之后，核心板蓝色电源指示灯、扩展板上两个蓝色指示灯（VCC5V 和 VCC3V3）全亮，表示电源工作正常。
3. 其他使用方法同 4.1 节。

4 核心板外设

4.1 时钟

核心板上提供一个 50MHz 的时钟 Y2；同时提供一个 SMA 接口连接外部时钟以满足对时钟要求较高的设计。图 13 给出了时钟的连接图。时钟 Y2 的 FPGA 引脚为 N1，SMA 外接时钟引脚为 N2（表 1）。

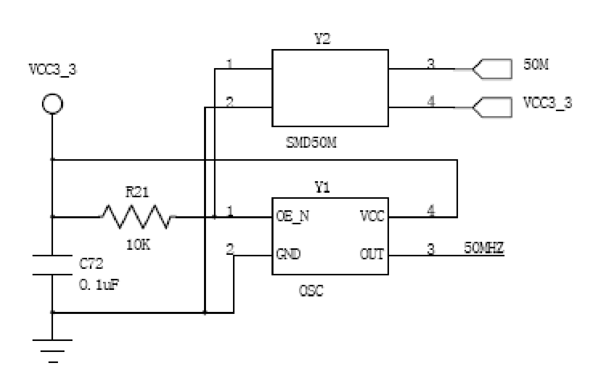


图 13　时钟连接图

表 1 时钟管脚映射

Y2_PIN 50MHz		
Y2 管脚编号	Y2 管脚名	FPGA 管脚编号
3	50M	N1
SMA		
SMA 管脚编号	SMA 管脚名	FPGA 管脚编号
3	50MHZ	N2

4.2 SRAM

核心板的 SRAM 采用了 ISSI 公司的异步 SRAM IS61LV25616，容量为 512k 字节，配置成 256k*16bit 方式。表 2 给出了 SRAM 的管脚映射。

表 2　SRAM 管脚映射（新版本）

SRAM IS61LV25616		
SRAM 管脚编号	SRAM 管脚名	FPGA 管脚编号
1	A0	AC9
2	A1	AD10
3	A2	AB10
4	A3	AC10
5	A4	AD11
6	CE	AC11
7	D0	AB12

8	D1	AD12
9	D2	AC12
10	D3	AA11
13	D4	AA12
14	D5	Y12
15	D6	AA10
16	D7	Y10
17	WE	W11
18	A5	Y11
19	A6	U12
20	A7	W12
21	A8	V13
22	A9	V14
23	A10	AF13
24	A11	AE13
25	A12	AE12
26	A13	AE11
27	A14	AF10
29	D8	AE10
30	D9	AF9
31	D10	AE9
32	D11	AF8
35	D12	AE8
36	D13	AF7
37	D14	AE7
38	D15	AA9
39	LB	AD8
40	UB	AC8
41	OE	AB8
42	A15	AD7
43	A16	AC7
44	A17	W8

4.3 FLASH

核心板上的 Flash 采用的是 Spansion 公司的 XOR FLASH S29GL064M，容量为 8Mbytes，可以升级到 32Mbytes。表 3 给出了 Flash 的管脚映射。

表 3 Flash 管脚映射

FLASH S29GL064M		
FLASH 管脚编号	FLASH 管脚名	FPGA 管脚编号
1	A24	AE24
2	A23	AE25
3	A16	AD25

4	A15	AC25
5	A14	AC26
6	A13	AB25
7	A12	AB26
8	A11	AA25
9	A10	AA26
10	A9	Y25
11	A20	Y26
12	A21	W25
13	WE#	W26
14	RESET#	V25
15	A22	V26
16	WP#/ACC	U25
17	RY/BY#	U26
18	A19	U24
19	A18	T25
20	A8	R25
21	A7	P24
22	A6	R24
23	A5	P23
24	A4	V21
25	A3	U22
26	A2	V20
31	A1	W24
32	CE#	AD24
34	OE#	AC23
35	DQ0	AB23
37	DQ1	AB24
39	DQ2	AA23
41	DQ3	AA24
44	DQ4	Y21
46	DQ5	Y22
48	DQ6	Y24
50	DQ7	Y23
51	DQ15/A-0	W21
53	BYTE#	W23
54	A17	V22

4.4 SDRAM

SDRAM 采用的是韩国现代公司的 HY57V561620，容量为 32M 字节。表 4 给出了 SDRAM 的管脚映射。

表 4 SDRAM 管脚映射

SDRAM HY57V561620

SDRAM 管脚编号	SDRAM 管脚名	FPGA 管脚编号
2	DQ0	L3
4	DQ1	K3
5	DQ2	F7
7	DQ3	F3
8	DQ4	G3
10	DQ5	F6
11	DQ6	F1
13	DQ7	H3
15	LDQM	G5
16	WE_N	J4
17	CAS_N	K5
18	RAS_N	L6
19	CS_N	L7
20	BA0	M4
21	BA1	L9
22	A10/AP	L10
23	A0	N9
24	A1	P9
25	A2	M3
26	A3	M2
29	A4	B3
30	A5	C3
31	A6	B2
32	A7	C2
33	A8	D2
34	A9	D1
35	A11	E2
36	A12	E1
37	CKE	F2
38	CLK	E5
39	UDQM	G2
42	DQ8	G1
44	DQ9	H2
45	DQ10	H1
47	DQ11	J2
48	DQ12	J1
50	DQ13	K2
51	DQ14	K1
53	DQ15	L2

4.5　串口

核心板串口采用了 MAX323C 串口芯片，并提供一 3 针的 RS232 接口（J6，图 14），G 是地线，TX 是发送线，RX 是接收线。核心板上有两个 LED 灯 D（D2，D3）指示收发状态。D3 闪烁表示有数据接收，D2 闪烁表示有数据发送。表 5 给出了串口的管脚映射。在连接串口线时，串口线上的黑色为地线(G)。

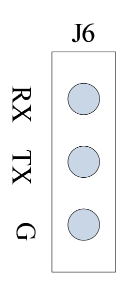


图 14　串口接口

表 5　串口管脚映射

UART MAX3232C		
MAX3232C 编号(TOP)	MAX3232C 管脚名	FPGA 管脚编号
11	UART_TXD	T21
12	UART_RXD	T20

4.6　以太网口

以太网芯片采用的是 Davicom 公司的 DM9000A，该芯片是 10/100 自适应以太网芯片，包括了 MAC 和 PHY 层的功能。该芯片既有通用处理器接口，也有 MII 的 PHY 层接口，本系统的设计中 FPGA 接连的是通用处理器接口。表 6 给出了以太网芯片管脚映射。

表 6　以太网芯片管脚映射

Ethernet DM9000A		
ETHERNET 管脚编号	ETHERNET 管脚名	FPGA 管脚编号
10	SD7	B4
11	SD6	C4
12	SD5	A4
13	SD4	A5
14	SD3	B5
16	SD2	A6
17	SD1	C6
18	SD0	B6
22	SD15	A7
24	SD14	B7
25	SD13	C7
26	SD12	A8
27	SD11	C8
28	SD10	B8

29	SD9	A9
31	SD8	B9
32	CMD	C9
34	INT	A10
35	IOR#	B10
36	IOW#	B11
37	CS#	C10
40	PWRES#	B12
44	X1(25MHZ)	F12

4.7 70*6 高速连接器

核心板的反面有 6 个 70 管脚的高速连接器（J1~J5、J7），可支持 100MHz 的高速信号，管脚说明参看附件，连接器管脚排序如图 15 所示。在附件原理图中管脚的排序是竖排的，逻辑图中只是逻辑符号，不是实际排序。

图 15 插座管脚排序

图 16 给出了核心版上各插座的位置（从反面看）。

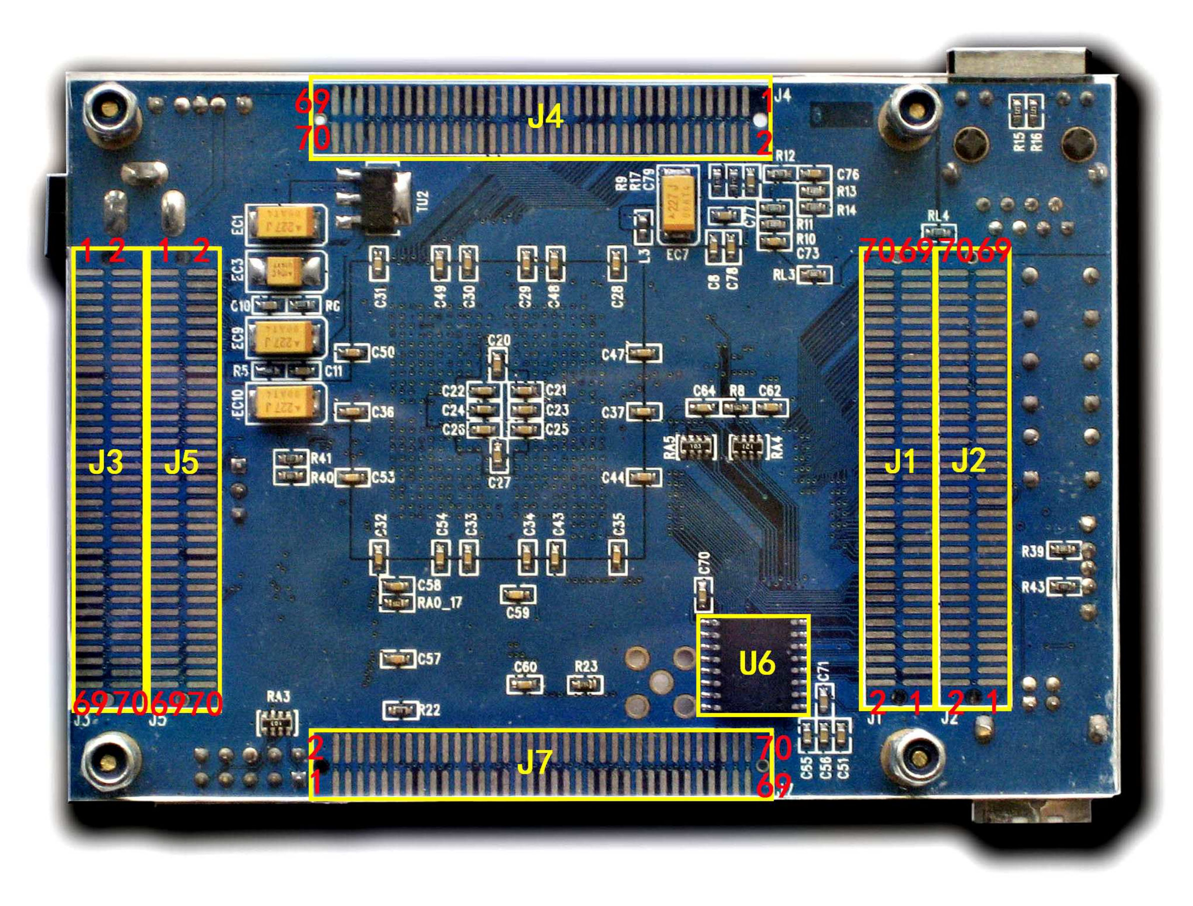


图 16　插座在核心板上的位置

4.8　LED 灯

核心板上有 8 个 LED 灯（LED7~LED0），相应的 FPGA 管脚拉低，LED 灯发光（图 17）。表 7 给出了 LED 的管脚映射。

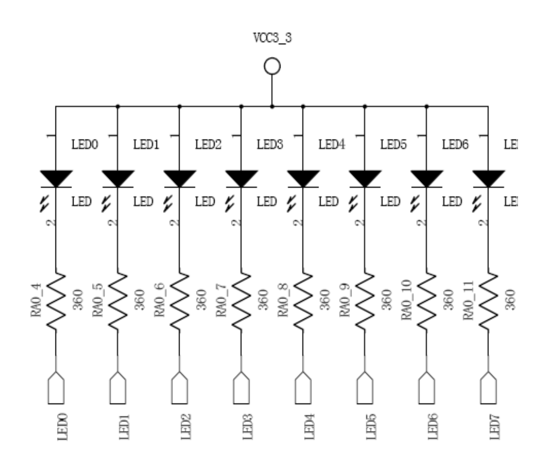


图 17　LED 灯连接图

表 7　LED 管脚映射

LED		
LED 编号	LED 管脚编号	FPGA 管脚编号
LED0	2	G6

LED1	2	G4
LED2	2	K6
LED3	2	H6
LED4	2	J3
LED5	2	J6
LED6	2	J7
LED7	2	K8

4.9 按钮

核心板上有三个按钮：RESET、KEY0 和 KEY1，三个按钮都连到普通 I/O 上，既可以当作普通 I/O 用，也可以作为 RESET 使用。按钮按下去接通低点平，缺省是高电平（图 18）。表 8 给出了按钮的管脚映射。

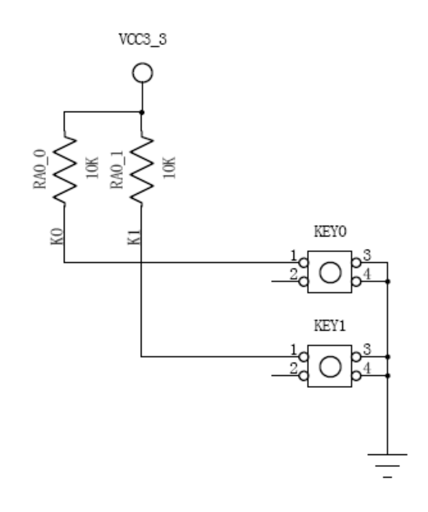


图 18 按钮连接图

表 8 按钮管脚映射

KEY		
按键编号	KEY 管脚编号	FPGA 管脚编号
KEY0	1, 2	F4
KEY1	1	L4
RESET	3	C5

5 扩展板外设

5.1 A/D

A/D 芯片采用的是 AD 公司的 9203，位宽为 10 比特，最高支持 40MHz 的采样率。芯片采用射频变压器交流耦合的方式送至 AD9203 的输入端。射频变压器 ADT1-1WT 的 3dB 带宽是 0.4~800MHz。AD9203 中频采样带宽可达 130MHz。板子上有个跳线（S1）可以设置参考电压，如果 S1 中间管腿和 AGND 连在一起，则参考电压是 1V；如果 S1 中间管腿和 VREF 连在一起，则参考电压是 0.5V。

表 9　FPGA 和 AD 的接口

AD9203				
AD Pin Num	AD Pin Name	FPGA Pin	扩展板连接器管脚	核心板连接器管脚
3	D0	PIN_E26	J12_43	J5_44
4	D1	PIN_E25	J12_44	J5_43
5	D2	PIN_F26	J12_45	J5_46
6	D3	PIN_F25	J12_46	J5_45
7	D4	PIN_G25	J12_48	J5_47
15	CLK	PIN_D26	J12_41	J5_42
8	D5	PIN_P17	J13_47	J3_48
9	D6	PIN_N24	J13_48	J3_47
10	D7	PIN_R17	J13_49	J3_50
11	D8	PIN_T17	J13_50	J3_49
12	D9	PIN_R19	J13_51	J3_52

5.2　D/A

D/A 芯片采用的是 TI 公司的 DAC904，转换速率达 165M 符号/秒，位宽为 14 比特。

表 10　FPGA 和 DA 的接口

DAC904				
DAC Pin Num	DAC Pin Num	FPGA Pin	J13(扩展板)	J3（核心板）
14	D0	PIN_M19	70	69
13	D1	PIN_P18	69	70
12	D2	PIN_L19	68	67
11	D3	PIN_M20	67	68
10	D4	PIN_V24	66	65
9	D5	PIN_V23	65	66
8	D6	PIN_U23	64	63
7	D7	PIN_T18	63	64
6	D8	PIN_U21	61	62
5	D9	PIN_T19	60	59
4	D10	PIN_T24	59	60
3	D11	PIN_T23	58	57
2	D12	PIN_R20	57	58
1	D13	PIN_T22	56	55
28	CLK	PIN_U20	53	54

5.3　4*4 扫描按键

扩展板上有一个 4*4 的扫描键盘（S3~S16）（图 19），共有 8 个引脚和 FPGA 的 IO 相连（表 11）。

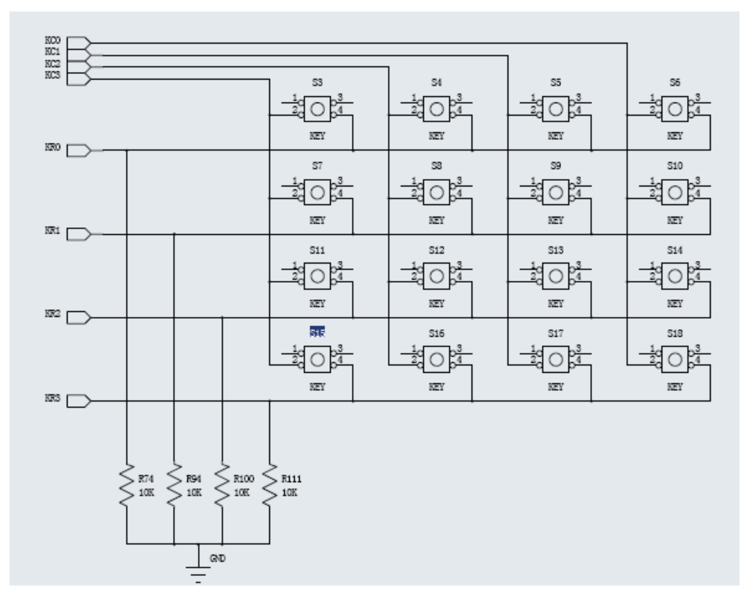


图 19　4*4 扫描按键示意图

表 11　扫描键盘管脚映射

KEY				
按键编号	按键管脚	FPGA 管脚	J16（扩展板）	J7（核心板）
S3	2	PIN_W15	61	62
	4	PIN_V10	70	69
S4	2	PIN_Y14	63	64
	4	PIN_V10	70	69
S5	2	PIN_AA14	64	63
	4	PIN_V10	70	69
S6	2	PIN_W10	67	68
	4	PIN_V10	70	69
S7	2	PIN_W15	61	62
	4	PIN_V9	69	70
S8	2	PIN_Y14	63	64
	4	PIN_V9	69	70
S9	2	PIN_AA14	64	63
	4	PIN_V9	69	70
S10	2	PIN_W10	67	68
	4	PIN_V9	69	70
S11	2	PIN_W15	61	62
	4	PIN_V11	68	67
S12	2	PIN_Y14	63	64
	4	PIN_V11	68	67
S13	2	PIN_AA14	64	63
	4	PIN_V11	68	67
S14	2	PIN_W10	67	68
	4	PIN_V11	68	67
S15	2	PIN_W15	61	62

	4	PIN_Y15	59	60
S16	2	PIN_Y14	63	64
	4	PIN_Y15	59	60
S17	2	PIN_AA14	64	63
	4	PIN_Y15	59	60
S18	2	PIN_W10	67	68
	4	PIN_Y15	59	60

扫描键盘的工作原理如下（可参看《实验指示书》的实验）：

KR0-KR3 为行线输入端，初始化为带下拉电阻的输入口。

KC0-KC3 为列线输出端，初始化为输出高电平。

表 12 行列端管脚映射

行列信号	FPGA 管脚
KC0	PIN_W10
KC1	PIN_AA14
KC2	PIN_Y14
KC3	PIN_W15
KR0	PIN_V9
KR1	PIN_V10
KR2	PIN_V11
KR3	PIN_Y15

键盘扫描方法：首先由输出口 KC0-KC3 向所有的列线输出高电平，读取各行线 KR0-KR3 的状态。若行线状态全为低电平，则表明无键按下，若有高电平，则表明有键按下。若有键按下，先进行消抖，消抖结束，再对每一列进行扫描，即：KC0-KC3 四条列扫描线分别输出高电平，在每次输出高电平期间，读取各行线 KR0-KR3 的状态，确定键值。例如：KC0-KC3 输出“0100”，此时若读取 KR0-KR3 的状态为“1000”，则表明按键 S5 被按下。

在按键闭合和打开的瞬间会产生许多尖脉冲，持续时间约几毫秒到几十毫秒。当检测到按键被按下后，便开始读取键值，可以每隔 M 毫秒读一次键值，直到连续 N 次读取的键值完全相同，则认为抖动已经消除。消抖时间为 $M\text{ms}\times N$，例如 M 取 10 毫秒，N 取 5。

5.4 拨动开关

扩展板有 10 个拨动开关（图 20），向下是相应管脚是高电平，向上拨相应管脚是低电平（图 21）。表 13 为拨动开关管脚映射。

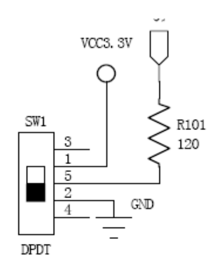


图 20 波动开关连接图

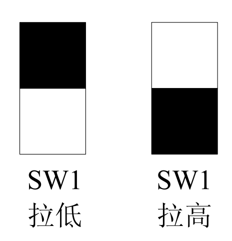


图 21　波动开关示意图（黑色表示开关位置）

表 13　波动开关管脚映射

SW				
SW	SW Pin	FPGA	J16 Pin（扩展板）	J7 Pin（核心板）
SW1	2	PIN_AB15	60	59
SW2	2	PIN_AA15	57	58
SW3	2	PIN_W16	58	57
SW4	2	PIN_AA16	55	56
SW5	2	PIN_AD15	49	50
SW6	2	PIN_Y16	56	55
SW7	2	PIN_U17	53	54
SW8	2	PIN_V17	54	53
SW9	2	PIN_W17	51	52
SW10	2	PIN_Y18	50	49

5.5　LED 灯

扩展板有 16 个 LED 灯（D21~D36），低电平发关（图 22），表 14 给出了 LED 的管脚映射。

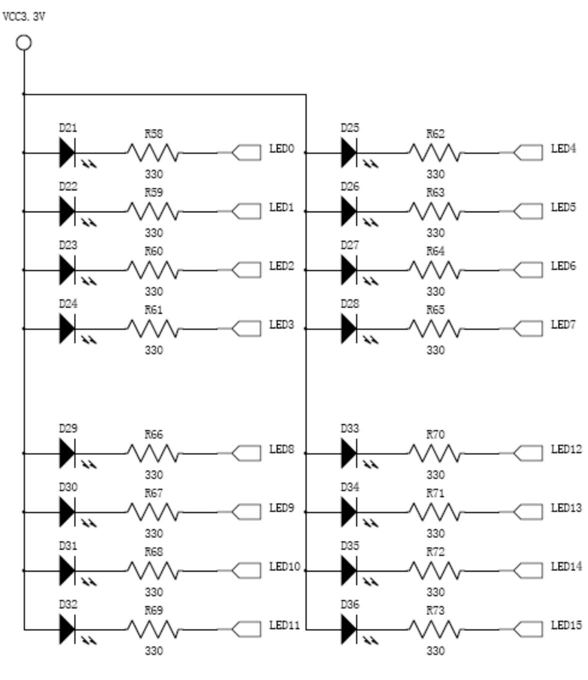


图 22　LED 灯连接图

表 14　LED 管脚映射

LED			
LED Num	FPGA Pin	J16(扩展板）	J7（核心板）
LED0	PIN_V18	47	
LED1	PIN_U18	48	48
LED2	PIN_AE15	44	47
LED3	PIN_AE16	41	43
LED4	PIN_AF17	40	42
LED5	PIN_AE18	37	39
LED6	PIN_AE19	35	38
LED7	PIN_AF19	36	36
LED8	PIN_AF20	34	35
LED9	PIN_AE21	31	33
LED10	PIN_AF21	29	32
LED11	PIN_AE23	28	30
LED12	PIN_AA13	26	27
LED13	PIN_AF23	25	25
LED14	PIN_Y13	23	26
LED15	PIN_AC16	20	24

5.6　扩展接口

扩展板上有两个 40 管腿的扩展接口（J17 和 J18），其管脚定义见附件。

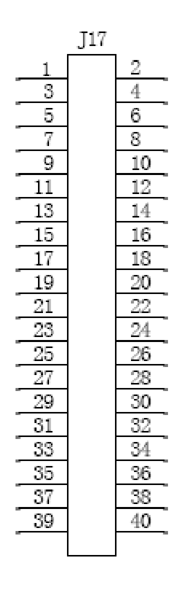


图 23　扩展接口管脚排列

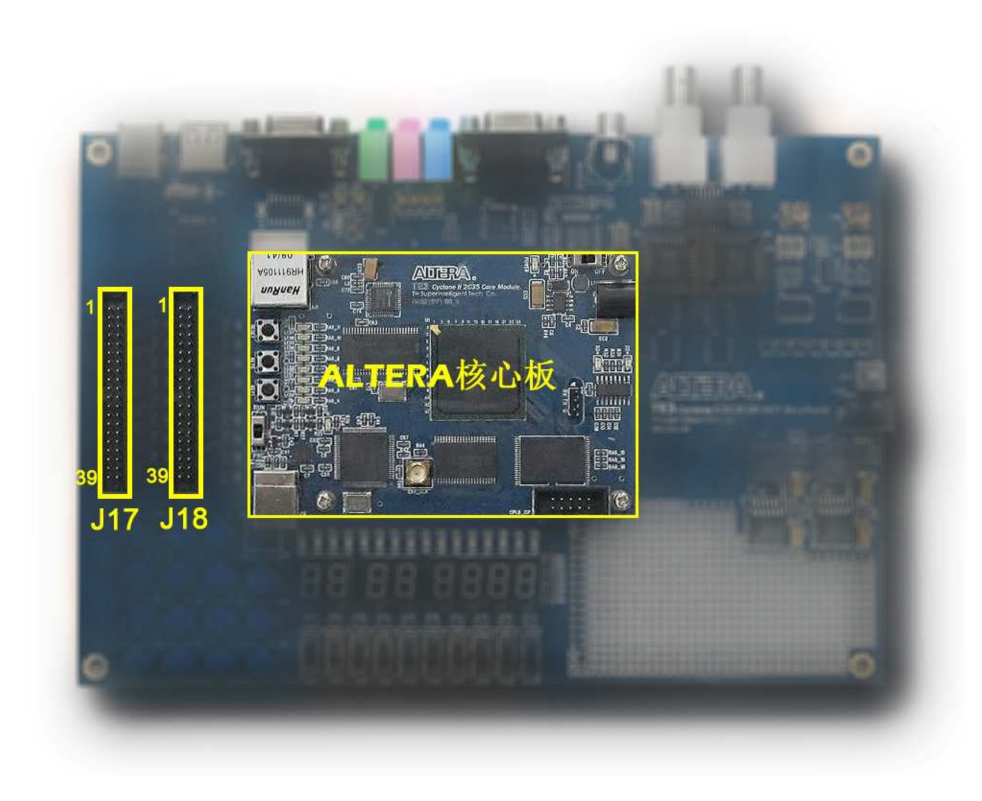


图 24　扩展接口在扩展板上的位置

表 15　J17 管脚连接

J17 管脚连接			
J17	FPGA	J17	FPGA
1	PIN_R5	2	PIN_J10
3	PIN_T8	4	PIN_U6
5	PIN_T10	6	PIN_U9
7	PIN_P4	8	PIN_R3
9	PIN_U4	10	PIN_T4
11	VCC5V	12	GND
13	PIN_V5	14	PIN_V7
15	PIN_Y3	16	PIN_W3
17	PIN_Y5	18	PIN_AA5
19	PIN_AD3	20	PIN_AB4
21	PIN_AA6	22	PIN_T2
23	PIN_V2	24	PIN_U2
25	PIN_AD4	26	PIN_W2
27	PIN_AC5	28	PIN_AE4
29	VCC3.3V	30	GND
31	PIN_AA1	32	PIN_AC6
33	PIN_AB1	34	PIN_AD5
35	PIN_AE5	36	PIN_AF4
37	PIN_AD6	38	PIN_AF5
39	PIN_AF6	40	PIN_AE6

表 16　J18 管脚连接

J18 管脚连接			
J18	FGPA	J18 Pin	FPGA Pin
1	PIN_J14	2	PIN_J13
3	PIN_T9	4	PIN_R6
5	PIN_T7	6	PIN_U5
7	PIN_V6	8	PIN_U7
9	PIN_U10	10	PIN_P3
11	VCC5V	12	GND
13	PIN_T6	14	PIN_R4
15	PIN_T3	16	PIN_U3
17	PIN_V3	18	PIN_V4
19	PIN_W4	20	PIN_W6
21	PIN_AA3	22	PIN_Y4
23	PIN_AA4	24	PIN_AB3
25	PIN_AE3	26	PIN_AC3
27	PIN_AA7	28	PIN_R2
29	VCC3.3V	30	GND
31	PIN_V1	32	PIN_U1
33	PIN_W1	34	PIN_Y1
35	PIN_AB2	36	PIN_AA2
37	PIN_AC2	38	PIN_AC1
39	PIN_AD2	40	PIN_AE2

5.7　3.0 寸 TFT 液晶彩色显示屏、触摸屏

3.0 寸液晶彩色显示屏控制芯片采用的是 ILI Technology Corp.的 ILI9326。表 17 给出了 TFT 屏的管脚映射。

表 17　显示屏控制芯片 ILI9326 管脚连接

TFT ILI9326				
ILI9326 管脚编号	ILI9326 管脚名	FPGA 管脚	扩展板连接器管脚	核心板连接器管脚
9	LCM_FMARK	PIN_AD23	J16.1	J7.2
10	LCM_Cs	PIN_K19	J12.63	J5.64
11	LCM_Rs	PIN_AC22	J16.2	J7.1
12	LCM_Wr	PIN_J21	J12.61	J5.62
14	LCM_Data0	PIN_AB21	J16.3	J7.4
15	LCM_Data1	PIN_L25	J12.60	J5.59
16	LCM_Data2	PIN_AD22	J16.4	J7.3
17	LCM_Data3	PIN_J20	J12.59	J5.60
18	LCM_Data4	PIN_AD21	J16.6	J7.5
19	LCM_Data5	PIN_K26	J12.58	J5.57
20	LCM_Data6	PIN_AC21	J16.7	J7.8

21	LCM_Data7	PIN_K25	J12.56	J5.55
22	LCM_Data8	PIN_AB20	J16.8	J7.7
23	LCM_Data9	PIN_J26	J12.54	J5.53
24	LCM_Data10	PIN_AC20	J16.9	J7.10
25	LCM_Data11	PIN_M25	J12.53	J5.54
26	LCM_Data12	PIN_AC19	J16.10	J7.9
27	LCM_Data13	PIN_J25	J12.51	J5.52
28	LCM_Data14	PIN_H26	J12.49	J5.50
29	LCM_Data15	PIN_H25	J12.50	J5.49
30	LCM_RST	PIN_G26	J12.47	J5.48

而是用的触摸屏采用的是 Texas Instruments 的 TSC2046。表 18 给出了触摸屏 TSC2046 的管脚映射。

表 18　显示屏控制芯片 ILI9326 管脚连接

触摸屏 TSC2046				
TSC2046 管脚编号	TSC2046 管脚名	FPGA 管脚	扩展板连接器管脚	核心板连接器管脚
1	TP_Busy	PIN_K21	J12.64	J5.63
2	TP_DI	PIN_K4	J10.29	J2.30
3	TP_EN	PIN_K18	J12.66	J5.65
4	TP_CLK	PIN_J11	J10.67	J2.68
15	TP_DET	PIN_L20	J12.65	J5.66
16	TP_DO	PIN_J5	J10.30	J2.29

5.8　USB 接口

USB 芯片采用的是 Philips 的 OTG 芯片 ISP1362，此芯片即可以当 Slave 用，也可以当 Master 使用。表 19 给出了 ISP 的管脚映射。

表 19　ISP1362 管脚映射

OTG ISP1362				
OTG 管脚编号	OTG 管脚名	FPGA 管脚	扩展板连接器管脚	核心板连接器管脚
61	A0	PIN_D6	J10_33	J2_34
62	A1	PIN_H8	J10_36	J2_35
63	D0	PIN_J8	J10_35	J2_36
64	D1	PIN_F10	J10_38	J2_37
2	D2	PIN_G9	J10_37	J2_38
3	D3	PIN_G10	J10_40	J2_39
5	D4	PIN_K9	J10_39	J2_40
6	D5	PIN_F11	J10_41	J2_42
7	D6	PIN_G11	J10_44	J2_43
8	D7	PIN_J9	J10_43	J2_44

10	D8	PIN_G12	J10_46	J2_45
11	D9	PIN_H10	J10_45	J2_46
12	D10	PIN_H12	J10_48	J2_47
13	D11	PIN_H11	J10_47	J2_48
15	D12	PIN_F9	J10_50	J2_49
16	D13	PIN_M5	J10_49	J2_50
17	D14	PIN_E8	J10_51	J2_52
18	D15	PIN_E10	J10_54	J2_53
20	RD#	PIN_E12	J10_53	J2_54
21	CS#	PIN_C11	J10_64	J2_63
22	WR#	PIN_D7	J10_56	J2_55
24	DREQ1	PIN_D5	J10_55	J2_56
25	DREQ2	PIN_D9	J10_58	J2_57
28	DACK1#	PIN_D8	J10_57	J2_58
29	DACK2#	PIN_D11	J10_60	J2_59
30	INT1	PIN_D10	J10_59	J2_60
31	INT2	PIN_D12	J10_61	J2_62
32	RESET#	PIN_C12	J10_66	J2_65
49	OTG_DM1	PIN_K7	J10_34	J2_33
50	OTG_DP1	PIN_H4	J10_31	J2_32

5.9 RS232C 串口

底板的 J9 是一个 9 针的 RS232 接口，D2 和 D3 分别是发送和接收的指示灯。表 20 给出了串口的管脚映射。

表 20 串口管脚映射

MAX3232C				
MAX 管脚编号	MAX 管脚名	FPGA 管脚	扩展板连接器管脚	扩展板连接器管脚
11	UART_TXD	PIN_K17	J8_63	J4_64
12	UART_RXD	PIN_J18	J8_64	J4_63

5.10 VGA 输出接口

VGA 输出接口芯片采用 AD 公司的 ADV7123 来 RGB 视频信号的 DA 变换。表 21 给出了 VGA 接口的管脚映射。

表 21 VGA 管脚映射

VGA ADV7123				
VGA 管脚编号	VGA 管脚名	FPGA 管脚	扩展板连接器管脚	核心板连接器管脚
39	R0	PIN_B20	J8_28	J4_27
40	R1	PIN_A20	J8_27	J4_28

41	R2	PIN_B21	J8_30	J4_29
42	R3	PIN_A21	J8_29	J4_30
43	R4	PIN_A22	J8_31	J4_32
44	R5	PIN_A23	J8_34	J4_33
45	R6	PIN_B22	J8_33	J4_34
46	R7	PIN_D14	J8_36	J4_35
47	R8	PIN_B23	J8_35	J4_36
48	R9	PIN_E15	J8_38	J4_37
1	G0	PIN_F14	J8_37	J4_38
2	G1	PIN_C15	J8_40	J4_39
3	G2	PIN_D15	J8_39	J4_40
4	G3	PIN_F15	J8_41	J4_42
5	G4	PIN_C16	J8_44	J4_43
6	G5	PIN_F16	J8_43	J4_44
7	G6	PIN_C17	J8_46	J4_45
8	G7	PIN_D16	J8_45	J4_46
9	G8	PIN_D18	J8_48	J4_47
10	G9	PIN_D17	J8_47	J4_48
14	B0	PIN_D19	J8_51	J4_52
15	B1	PIN_D20	J8_54	J4_53
16	B2	PIN_F18	J8_53	J4_54
17	B3	PIN_C21	J8_56	J4_55
18	B4	PIN_G18	J8_55	J4_56
19	B5	PIN_C22	J8_58	J4_57
20	B6	PIN_E20	J8_57	J4_58
21	B7	PIN_J17	J8_60	J4_59
22	B8	PIN_D21	J8_59	J4_60
23	B9	PIN_C23	J8_61	J4_62
11	BLANK#	PIN_C19	J8_50	J4_49
12	SYNC#	PIN_E18	J8_49	J4_50
24	C1K	PIN_F20	J12_1	J5_2

5.11 AV 输入接口

AV 输入采用 RCA 接口，接口芯片采用的是 AD 公司的 ADV7181，完成输入信号的 AD 变换。表 22 给出了 ADV 的管脚映射。

表 22　AV 管脚映射

ADV7181B				
ADV 管脚编号	ADV 管脚名	FPGA 管脚	扩展板连接器管脚	核心板连接器管脚
51	RESET#	PIN_F21	J12_2	J5_1
54	SCLK	PIN_E22	J12_4	J5_3

53	SDATA	PIN_C24	J12_3	J5_4
8	P8	PIN_H24	J12_15	J5_16
7	P9	PIN_G24	J12_14	J5_13
6	P10	PIN_G22	J12_13	J5_14
5	P11	PIN_G23	J12_11	J5_12
62	P12	PIN_G21	J12_8	J5_7
61	P13	PIN_E24	J12_7	J5_8
60	P14	PIN_E23	J12_6	J5_5
59	P15	PIN_D23	J12_5	J5_6
2	HS	PIN_F23	J12_10	J5_9
64	VS	PIN_F24	J12_9	J5_10
20	LLC	PIN_P25	J13_54	J3_53

5.12 数码管

扩展板有 8 个共阳八段数码管，由于采用独立控制的方式需要大量的 I/O 接口，因此数码管采用了扫描的连接方式（图 25）。人眼具有视觉暂留的特性，其影像在人眼的视网膜上仍旧会残存一段时间，因此当一个事物周期性的闪现在眼前，且其重复频率足够高时，这个事物的影像就会一直留在人眼的视网膜上，则在人眼看来这个事物就是连续存在的。对于每个显示器，如果让它每隔时间 T 闪现一次，而每次闪现时间为 t（满足 t<T），则当 T 足够小（例如 T=10ms 时），我们看这个显示器就像是连续显示的了。如果要同时显示 8 个数码管，则我们可以安排把每个周期 T 平均分成 8 段，每个显示器分别利用其中的一段时间进行显示（即 t=T/8）。

如果数码管的 TLx(x=0~7)端拉底，三极管导通，1、6 共阳两个腿拉高，这时候输入的数据从数码管显示出来。所以扫描的时候，通过不断地控制 TL 端来控制数据是否显示。

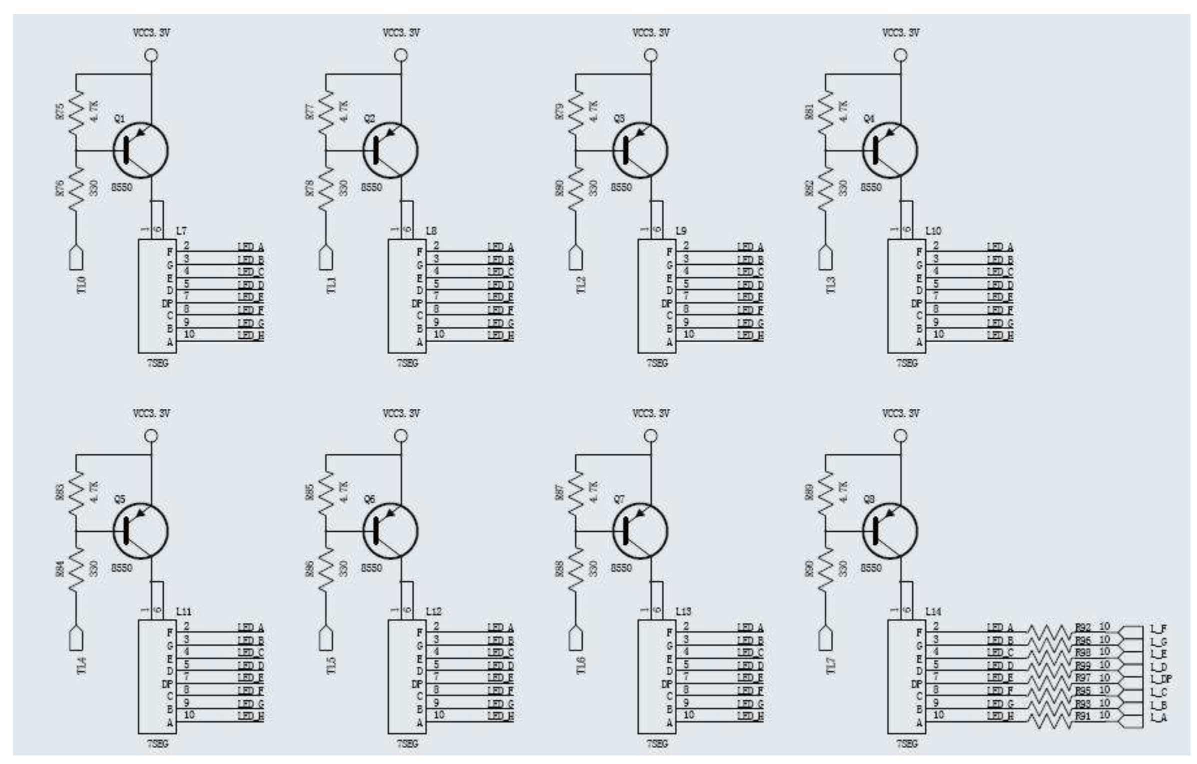


图 25　数码管连接图

表 23 和表 24 分别给出了数码管和控制端的管脚映射。

表 23　数码管管脚映射

数码管管脚映射				
管脚编号	管脚名	FPGA 管脚	扩展板连接器管脚	核心板连接器管脚
10	A	PIN_AD16	J16_19	J7_20
9	B	PIN_AC17	J16_17	J7_18
8	C	PIN_AB18	J16_15	J7_16
5	D	PIN_AD19	J16_11	J7_12
4	E	PIN_AC18	J16_14	J7_13
2	F	PIN_AA17	J16_18	J7_17
3	G	PIN_AD17	J16_16	J7_15
7	DP	PIN_AA18	J16_13	J7_14

表 24　数码管三极管管脚映射

数码管三极管管脚映射				
数码管编号	管脚	FPGA 管脚	扩展板连接器管脚	核心板连接器管脚
L7	TL0	PIN_W19	J16_45	J7_46
L8	TL1	PIN_AE17	J16_39	J7_40
L9	TL2	PIN_AF18	J16_38	J7_37
L10	TL3	PIN_AE20	J16_33	J7_34
L11	TL4	PIN_AE22	J16_30	J7_29
L12	TL5	PIN_AF22	J16_27	J7_28
L13	TL6	PIN_AC14	J16_24	J7_23
L14	TL7	PIN_AC15	J16_21	J7_22

5.13 音频接口

音频接口芯片采用的是飞利浦公司的 UDA1341TS。

J5: Audio Line out.

J4:Audio Line In.

J3:Audio Mic In.

表 25　数码管三极管管脚映射

Audio UDA1341TS				
音频管脚编号	音频管脚名	FPGA 管脚	扩展板连接器管脚	核心板连接器管脚
17	WORDSEL	PIN_F13	J8_2	J4_1
16	BITCLK	PIN_G13	J8_1	J4_2
15	L3_DATA	PIN_G15	J8_4	J4_3
12	SYSCLK	PIN_G14	J8_3	J4_4
18	DATAOUT	PIN_R7	J14_68	J1_67

19	DATAIN	PIN_P6	J14_67	J1_68
13	L3_MODE	PIN_R8	J14_70	J1_69
14	L3_CLK	PIN_P7	J14_69	J1_70

6　常用 EDA 软件使用

6.1　综合工具 Quartus II 的使用

Quartus II 是 Altera 公司针对其 FPGA 推出的集成开发环境。在这里通过一个名为 LED 的工程来简要地介绍 Quartus II 的设计流程，该工程能够实现核心板上的 LED 灯不断地来回闪烁。本文采用的是 Quartus II 7.0 版本，其他版本的使用方法类似。

6.1.1　工程建立

1）打开 Quartus II（【Windows 的开始】->【程序】-【Altera】->【Quartus II 7.0】->【Quartus II 7.0 (32-bit)】)，点击【File】->【New Project Wizard】选项，会出现如图 26 所示对话框，第一行用于指定工程路径，第二行用于指定工程名，第三行用于设定最高层设计名，在默认状态下跟工程名一致，用户可以根据自己喜好修改。 这里创建一个名为 LED 的工程。

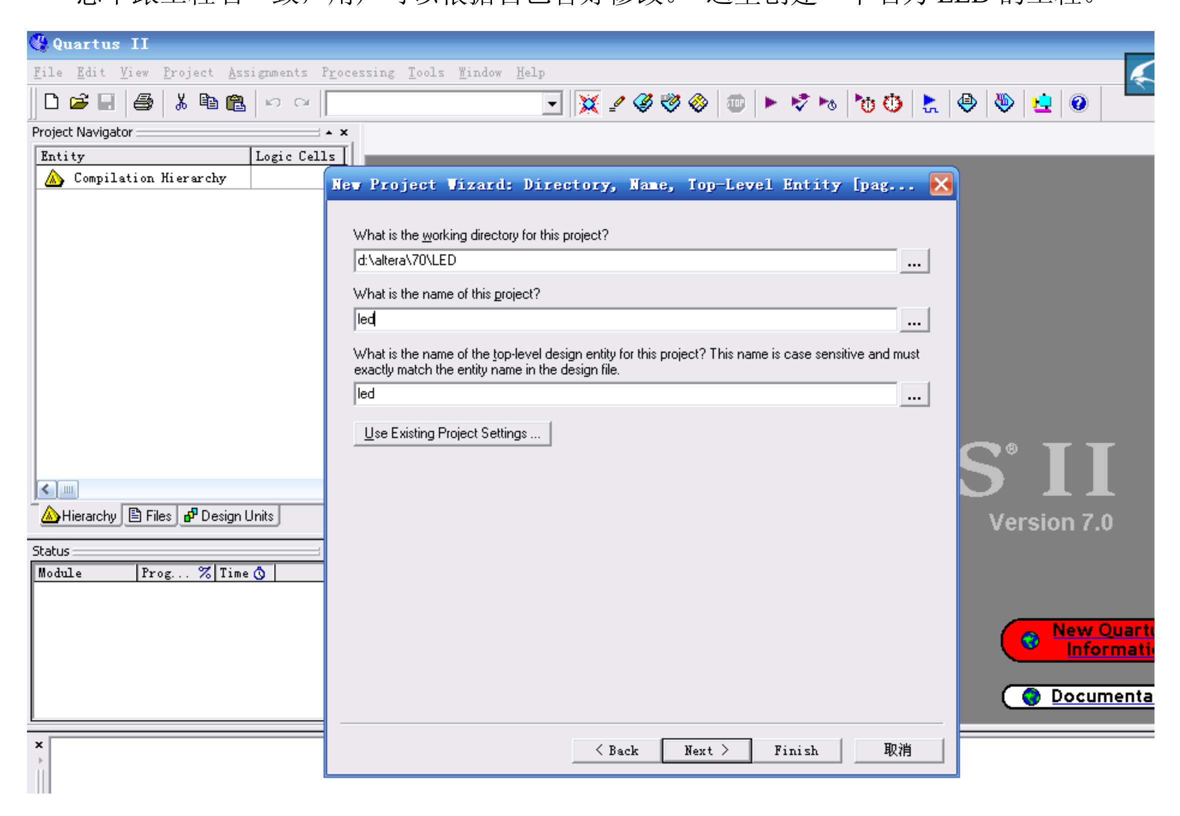


图 26　工程建立界面

2）点击【Next】，出现添加文件对话框，如图 27 所示。用户可以将已编辑好的设计文件添加到工程中。在本例中不添加任何文件，直接点击【Next】。

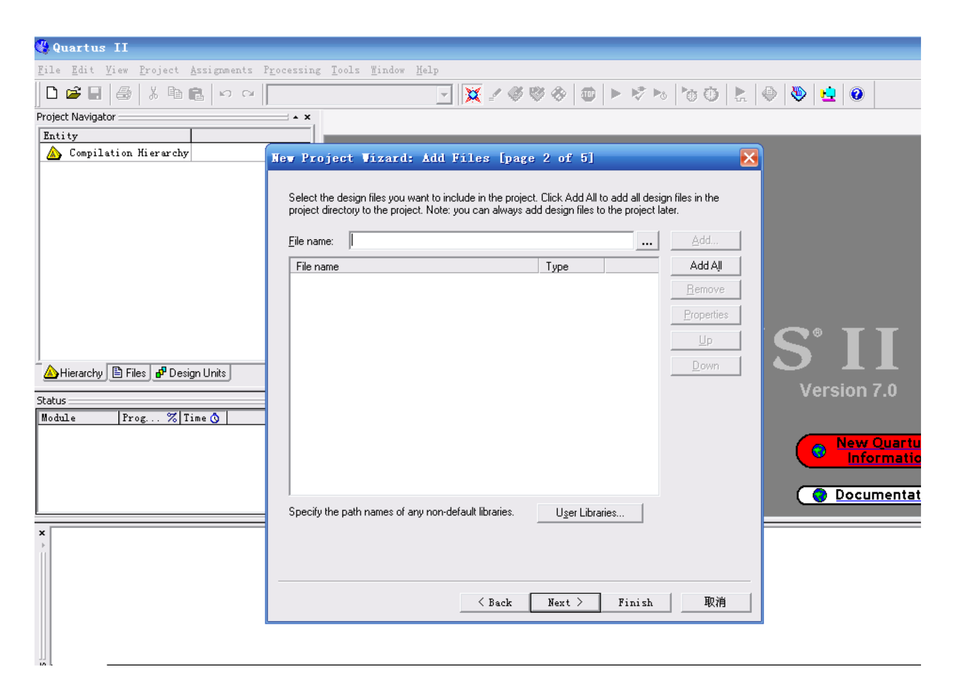


图 27　文件添加界面

3）点【Next】->【Family & device setting】窗口，此对话框用来选择目标器件，即所使用的 FPGA 型号。这个选项与最后综合生成的下载文件 sof 相关，这里选择 EP2C35F672C8。

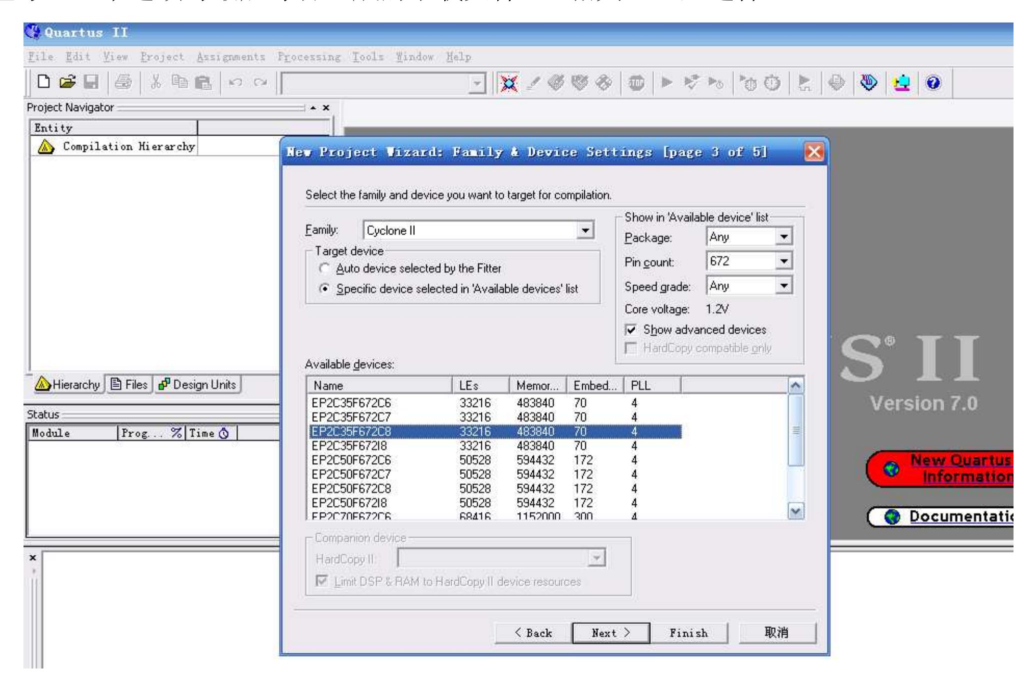


图 28　器件设置界面

4）点击【Next】，进入第三方工具选择窗口，用户可以在此选择其他公司的 EDA 工具，例如 Synopsys，MentorGraphic 等，在本例中不选用第三方工具。继续点击【Next】，进入 LED Summary 窗口，该窗口给出了所建 LED 项目的信息摘要。点击【Finish】，完成工程建立。

注意：上述第二步之后的设置选项在设置完毕后仍可在【Assignments】菜单下的【Settings】选项中修改。

6.1.2 建立设计

Quartus II 支持多种设计输入方式，可以在 Quartus II Block Editor 中建立原理图设计；或使用 Quartus II Text Editor 编辑 AHDL、Verilog HDL 或 VHDL 文件来进行设计；Quartus II 软件还支持采用 EDA 设计输入和综合工具生成的 EDIF 输入文件(.edf) 或 VQM 文件 (.vqm) 来建立设计。这里主要介绍在 Quartus II 中如何进行硬件语言设计和原理图设计。

1） 硬件语言设计

Quartus II 支持的硬件语言有 AHDL、Verilog HDL 或 VHDL，这里新建一个名为 led_core 的 Verilog HDL 文件。

a) 在主窗口下选择【File】->【New】，选择 Verilog HDL File 并点击【OK】就可以出现文本编辑框。

b) 对于初学者 Quartus II 提供了很多设计模板，在编辑框中点击右键选择【Insert Template】，对话框左边显示的 Quartus II 提供的模板类型，这里选择 Verilog HDL，在右边选择 Module Declaration，如图 29 所示。这时编辑框中会出现 Verilog HDL 模块声明的模板。

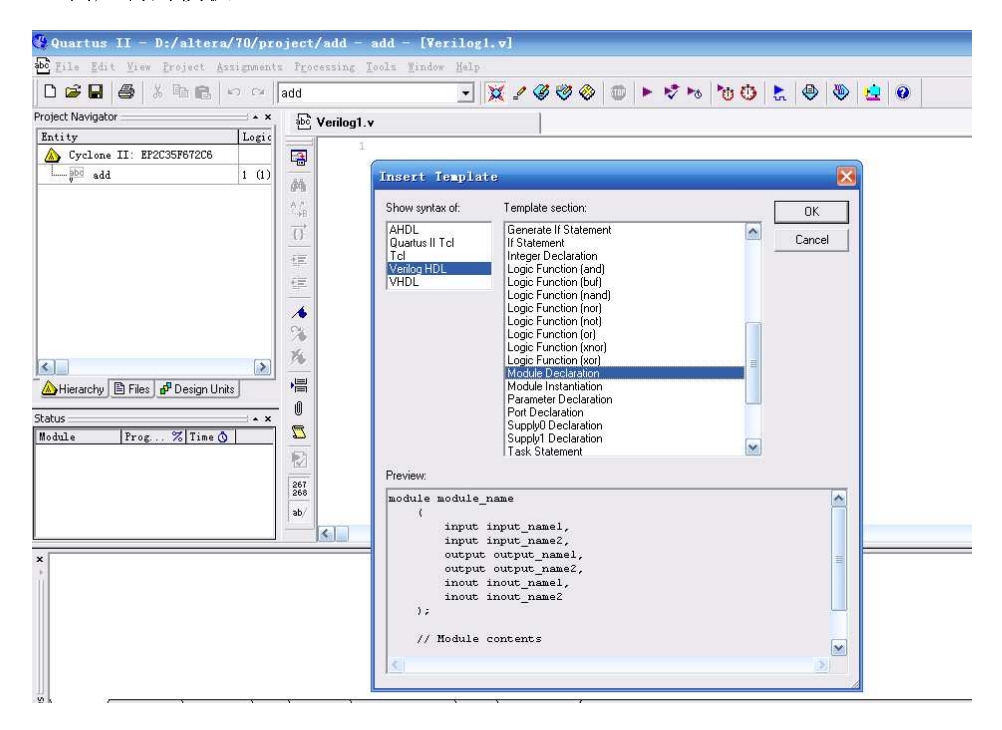


图 29 Verilog HDL 模板添加

c) 在文本框中输入 Verilog HDL 代码（图 30），并保存为 led_core.v 文件。

```
1  module led_core(clk, reset, LED);
2      input clk,reset;
3      output [7:0] LED;
4      reg [3:0] state;
5      reg [24:0]counter;
6      reg [7:0] LED;
7
8      always @(posedge clk)
9          if(!reset)
10             counter <= 25'b0;
11         else
12             counter <= counter + 1;
13
14     always @(posedge clk) begin
15         if(!reset)
16             state <= 4'b1000;
17         else
18             if (counter == 24'b0)
19                 if(state <= 4'b0111 && state >=4'b0000)
20                     state <= state - 1;
21                 else
22                     state <= 4'b0111;
23     end
```

图 30 Verilog HDL **代码输入界面**

d) 在 Quartus II 可以将硬件语言设计转换为原理图设计的符号单元，如图 31 所示。点击【File】->【Create/Update】->【Create Symbol Files For Current File】，将上一步的 led_core.v 文件设计转换为 Smbol 文件。

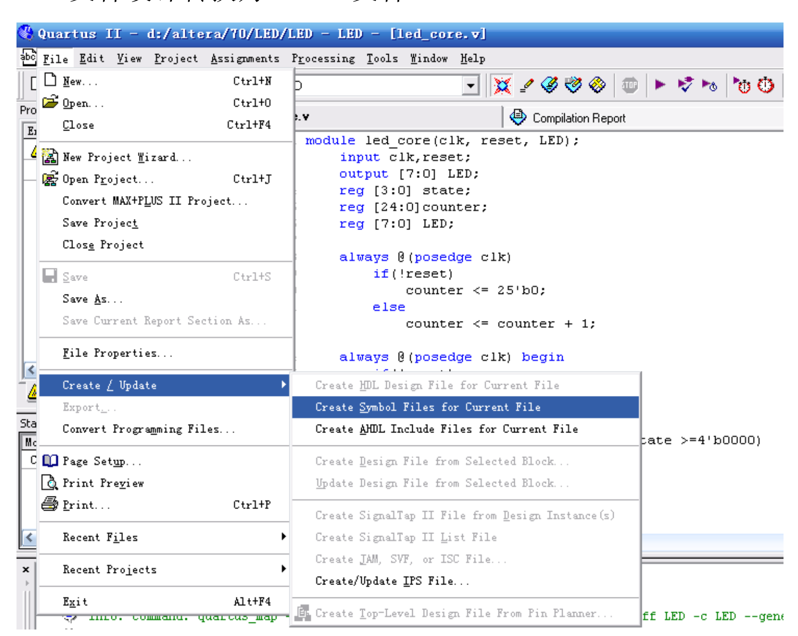


图 31 Verilog HDL **文件转化为** Symbol **文件**

2） 原理图设计

Quartus II Block Editor 用于以原理图和流程图的形式输入和编辑图形设计。每个原理图设计文件包含块和符号（Symbol），这些块和符号代表设计中的逻辑，Block Editor 将每个流程图、原理图或符号代表的设计逻辑融合到工程中。这里使用上一步生成的符号完成原理图设计。

a) 新建一个 Block Diagram/Schematic File.在编辑框中双击鼠标左键，会出现符号对话框，如图 32 所示。在左边 Project 文件夹下会选择 led_core，在右边显示出 led_core 的符号，点击【OK】后，可以将符号添加到设计中。

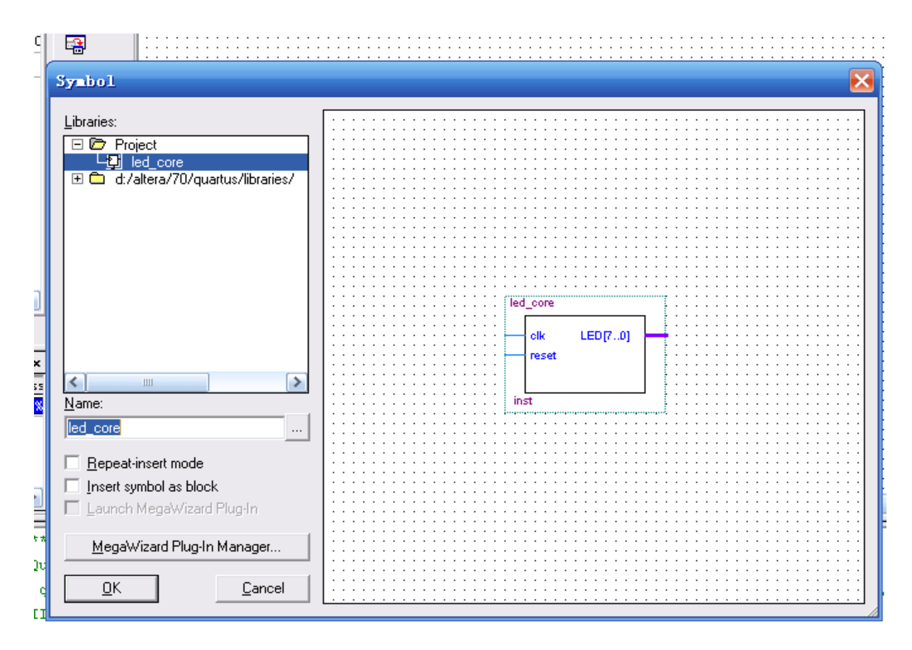


图 32　符号对话框界面

b） 接下来还需要添加输入输出口，在符号对话框中的 Name 中输入 input 这时会出现输入端口的符号。将此符号添加到原理图编辑框中并与 led_core 的输入端 clk 相连，如图 33 所示。

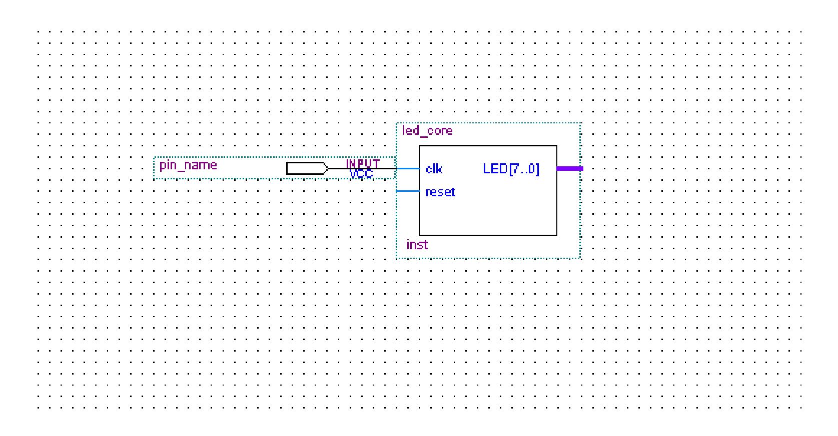


图 33　输入输出管脚添加

将上述的符号复制并与 reset 端口相连。按照与 input 相同的方式在 Symbol 对话框中找到 output 并与输出口 LED 相连。

c） 对上述输入输出口的名字进行修改，双击符号标识在 Pin name 下输入端口名。注意对于向量输出口的定义应该 LED[7..0]，而不是 LED[7:0]。

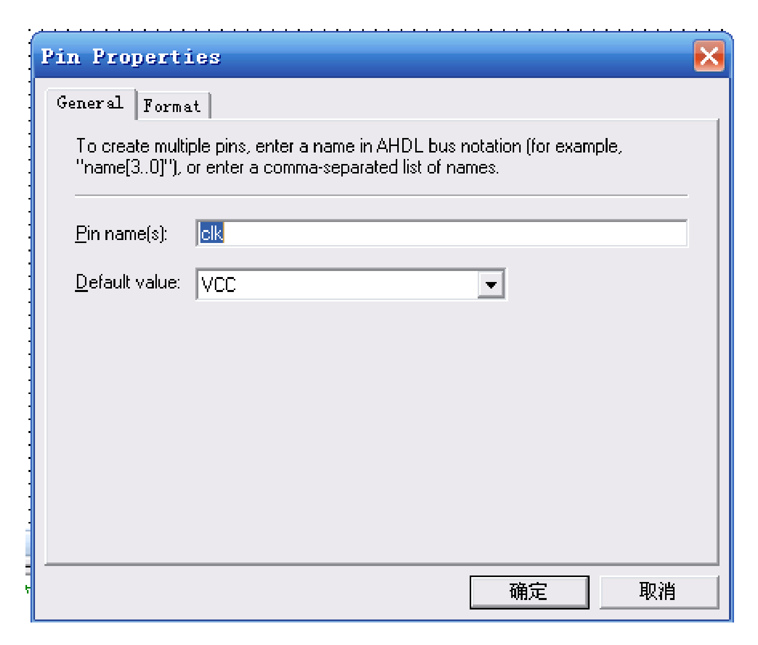


图 34　输入输出管脚属性设定

d) 至此，整个原理图设计完成，如图 35 所示，将该设计保存为顶层设计文件名 led。此文件名跟之前工程建立时定义的顶层文件名相同。

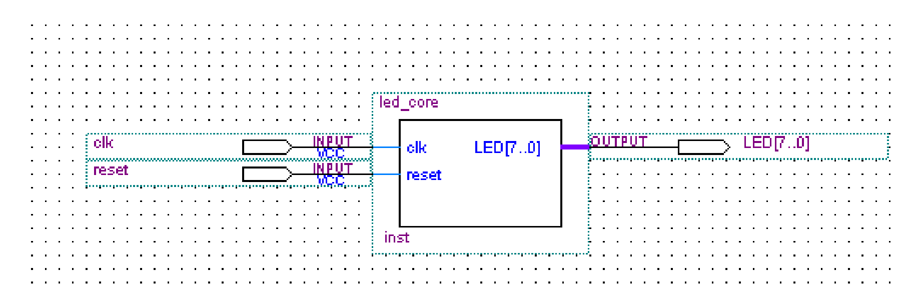


图 35　LED 原理图设计

6.1.3　分析与综合

选择 Quartus II 主窗口的【Processing】 ->【Starting】，可以看到 Quartus II 集成的分析综合工具，如分析查错，逻辑综合，布局布线，时序分析等，如图 36 所示。点击 Start Analysis & Synthesis 进行分析与综合。

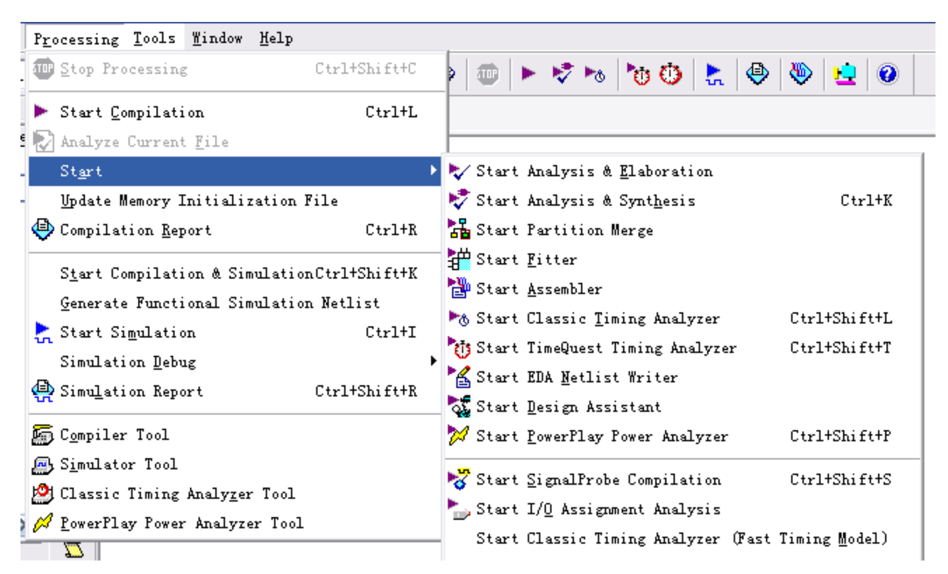


图 36　Quartus II 分析综合工具栏

在编译前，可以在 Settings 选项中设置一些参数使得编译器采取一些特别的综合和适配技术提高工程编译的速度，优化器件的资源利用率。这里，我们直接将上面的工程进行综合。

6.1.4　管脚分配

选择【Assignments】->【Pins】，进行这个工程的管脚分配，如图 37 所示，在 Location 栏里填写与输入输出对应的 Pin 脚。

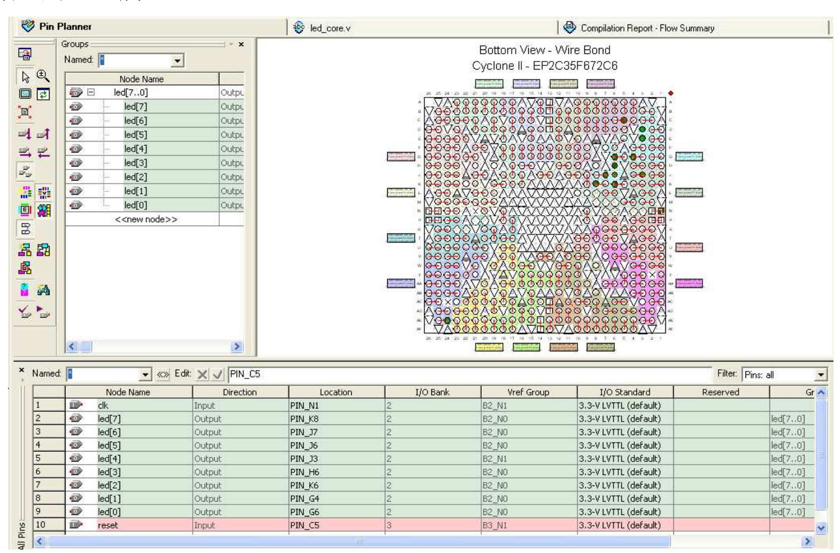


图 37　Quartus II 管脚分配界面

管脚分配的所有信息都被保存在工程所在文件夹的.qsf 文件中，我们可以直接编辑这个文件来修改管脚分配。

注意对于不使用的管脚，在 settings 对话框中对 Unused Pins 设置为 input tri-stated，如图 38 所示。

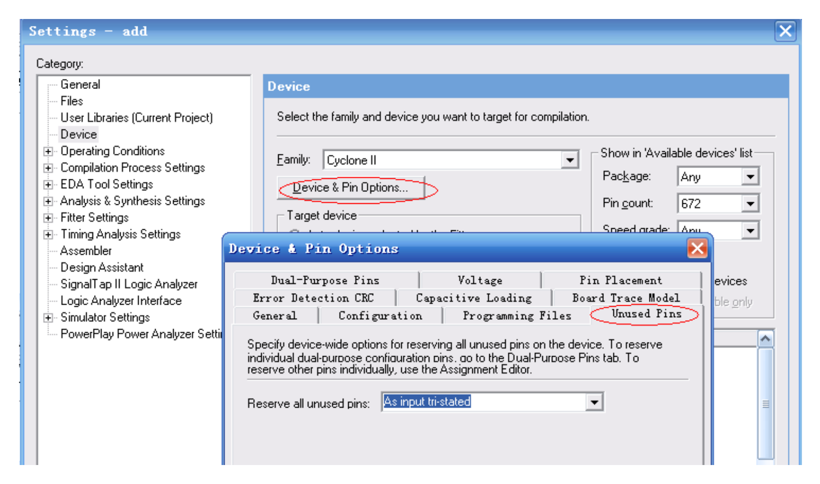


图 38　Unused Pins 设置界面

管脚分配完成后点，开始编译，编译成功后整个工程的软件设计部分全部完成，接下来将程序下载到目标板上并进行硬件调试。

6.1.5　程序下载

编译结束后，Quartus II 会生成两个下载程序，对于.sof 和.pof 文件。其中.sof 用于 JTAG

模式下载，这种方式程序将直接下载到 FPGA 中，断电后程序丢失。而.pof 则用于 AS 模式下载(Active Serial)，这种方式程序将直接下载到 EPCS 存储器中，断电后程序不会丢失，每次加电后 EPCS 中的程序会自动加载到 FPGA 中。具体下载方法如下：

1） 打开电源，将 PC 与目标板用下载线连接好。点击 Quartus II 中的 Programmer 键，在默认情况下会出现. sof 程序。点击 Hardware Setup 键，Quartus II 自动搜索 USB-Blaster，如果没有搜索成功，可能是没有安装驱动程序。在搜索到 USB-Blaster 后，在 Hardware Setup 中选上 USB-Blaster。在 Program/Configure 方框中打勾，下载模式选择 JTAG。将目标板上的下载模式开关拨在 RUN 状态，然后点击 Start 开始下载，如图 39 所示。

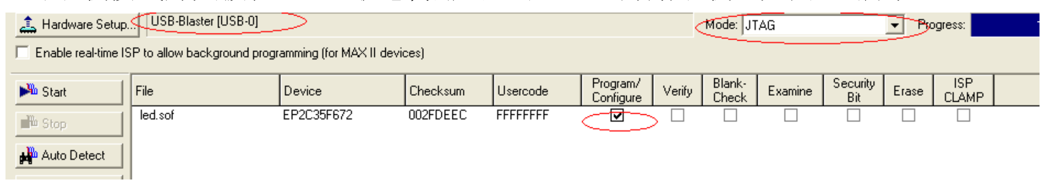


图 39　Jtag 方式程序下载界面

2） 如果需要下载. pof 文件，首先点击 Delete 选项将. sof 文件从 Programmer 中删除，然后点击 Add File 选择.pof 文件,将模式(Mode)改为 Active Serial，如图 40。注意此时要将目标板上的下载模式开关拨在 PRG 状态，然后点击 Start，程序被下载到 EPCS 中。

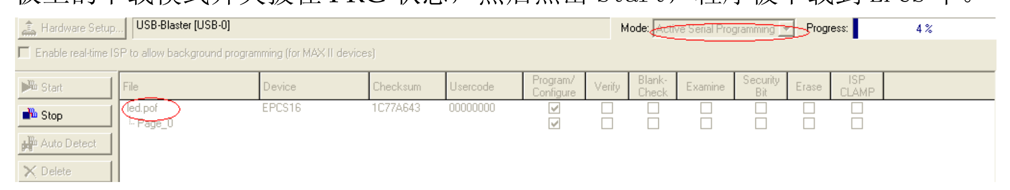


图 40　AS 方式程序下载界面

至此，整个完成了整个工程的建立，在目标板上可以看到 LED 灯循环亮起。

6.2　Signaltap II 的使用

Signaltap II 是 Quartus II 中集成的逻辑分析工具，可以对设计中的内部信号状态进行在线探测。在 FPGA 调试中起着重要作用。这里对它的使用进行简要介绍。

选择【Tools】->【Signaltap II Logic Analyzer】，出现如图 41 所示对话框。

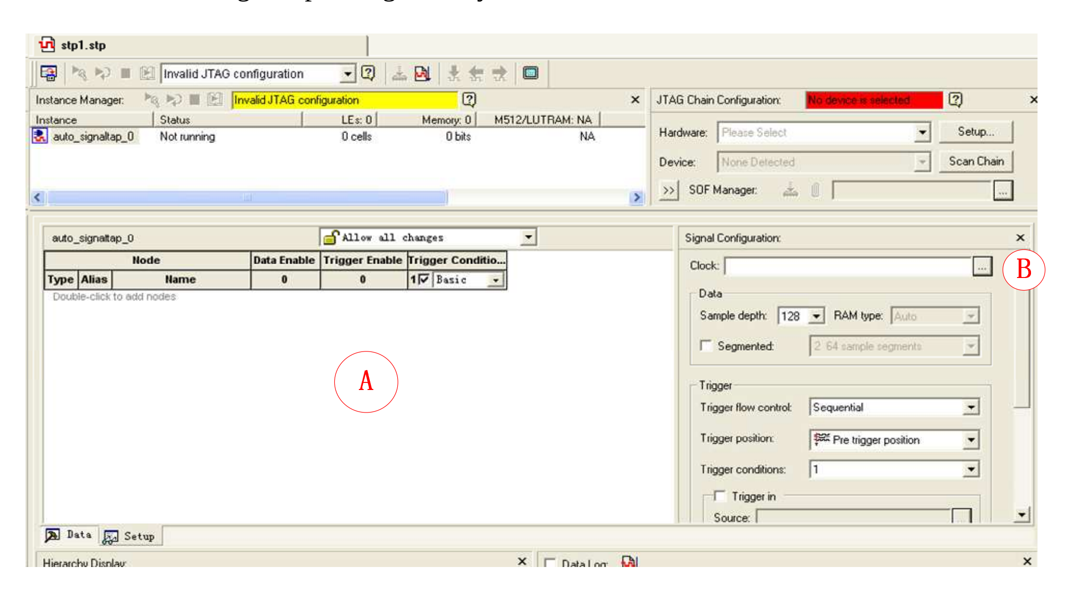


图 41　Signaltap II 界面

1.　signal configure 栏进行设置

首先设置触发时钟 Clock，点击图 41 中 B 处的按钮出现如图 42 所示对话框，选择【Filter】->【Post-fitting】，然后点击 List，Nodes Found 中出现所有的结点，选择 clk 点击并向右键，这时 clk 出现在右边的框内表示它被选中，点击【OK】，完成设置。

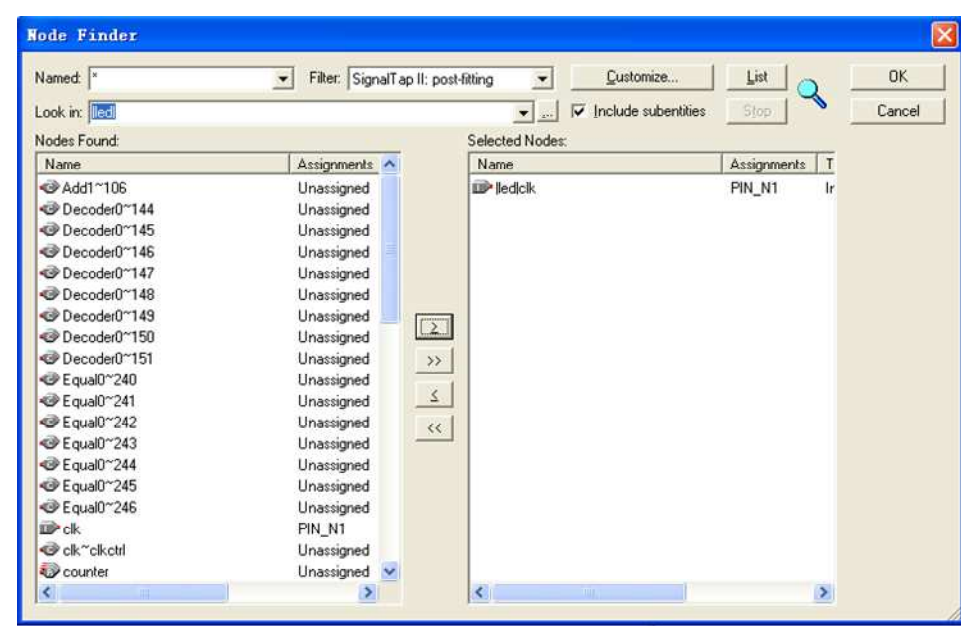


图 42　触发时钟设置

Sample depth 选项用来选择的采样的深度，可以根据需要进行设置，但是必须考虑到 FPGA 内部的存储器的大小，如果设置超出 FPGA 内部存储器的大小则该设置不能通过。

2. 观测变量的选择

在图 41 中 A 区域双击鼠标左键会出现与触发时钟类似的对话框（图 42），具体选择的方法一样，在这里我们选则内部逻辑变量 state 和 counter，点击【OK】后回到图 43。

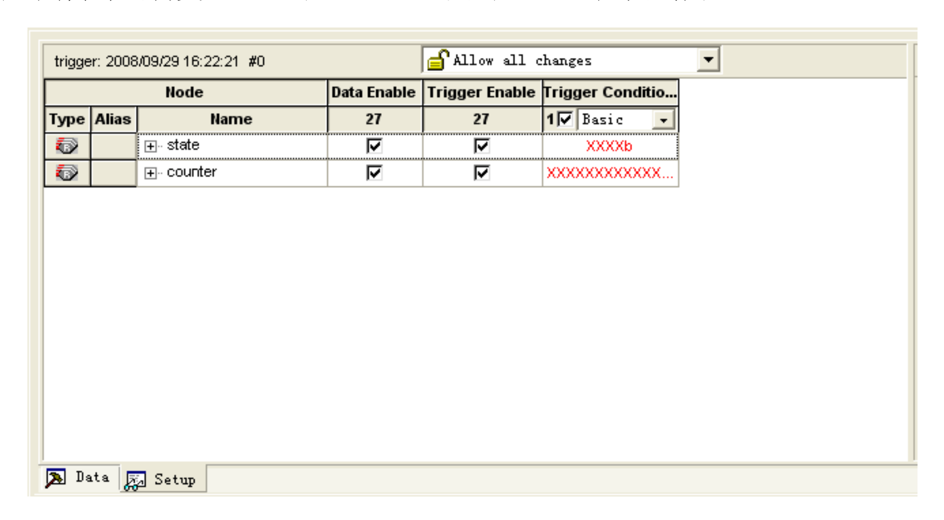


图 43 目标结点设置

将上述设置保存并对工程进行重新编译。

3. JTAG 设置

将程序下载到目标板中，对 Signaltap II 中的 JTAG Chain Configuration 栏进行设置。点击 Setup 选择 USB-Blaster，点击 Scan Chain 扫描出目标器件，最后 SOF Manager 中选择 led.sof，如图 44 所示。

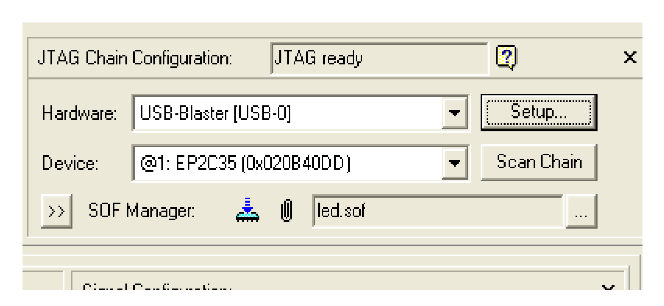


图 44 Jtag 连接设置

至此完成了 Signaltap 的所有设置。

4. 观察逻辑变量

点击 Instance Manager 栏中的键，这时便会出现 state 和 counter 两个变量的采样数据。如图 45 所示。

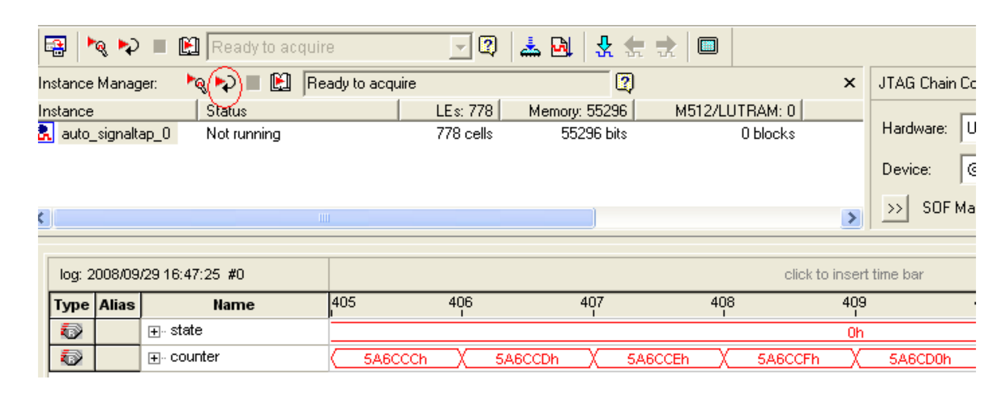


图 45　内部逻辑观察窗口

6.3　仿真工具 ModelSim 的使用

Modelsim 是 MenterGraphics 公司推出的一种简单易用，功能强大的逻辑仿真工具。这里对上面的 led_core.v 进行代码仿真来简要介绍 ModelSim 的使用。代码仿真需要的以下三个部分：

1. 被测试的设计代码，VHDL 语言或 Verilog 语言形式。
2. 测试激励代码，又称为 Test bench，模拟外部的输入，例如复位，时钟等，这些在仿真的时候只能通过 test bench 来产生。
3. 仿真模型/库：根据设计内调用的器件供应商提供的模块而定，如： Altera 提供的软核 PLL，Buffer 等。

ModelSim 代码仿真步骤如下。

6.3.1　ModelSim 仿真过程

1）建立工程

点击【File】->【New】->【Project】，这时会出现 Create Project 的窗口，如图 46 所示。在 Project Name 栏中填写项目名字，建议和你的顶层文件名字一致，这里设为 test。Project Location 指定工作目录，可通过 Browse 按钮来选择或改变。Default Library Name 可以采用工具默认的 work。

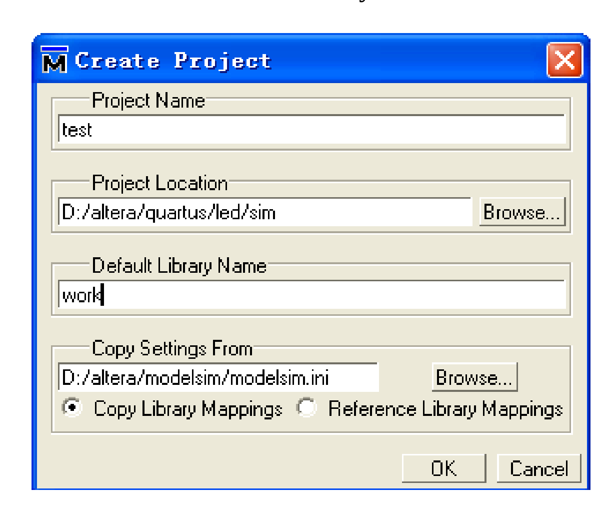


图 46　Modelsim 工程建立

2）添加文件

ModelSim 会自动弹出 LED Items to the project 窗口，给工程加入文件，如图 47 所示。

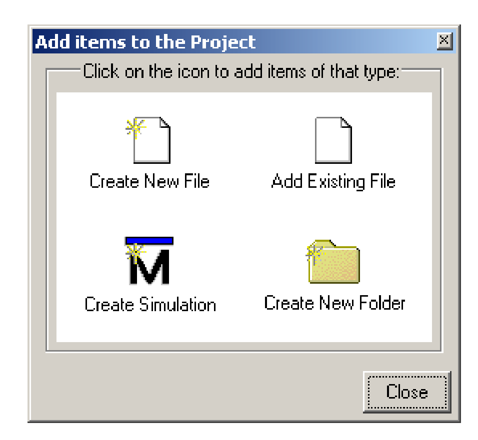


图 47　文件添加界面

选择 LED Exsiting File 后，根据相应提示将文件加到该 Project 中，这里加入两个文件被测试文件 led_core.v （图 48）和测试激励 test.v（图 49），具体程序如下。

```
`timescale 1ns / 1ps
 module led_core( clk, reset ,led );
  input clk, reset;
  output [7:0] led;
  reg [3:0] state;
  reg [22:0] counter;
  reg [7:0] led;

  always @(posedge clk)
      if(!reset)
          counter<= 23'b0;
      else
          counter<= counter+1;

  always @(posedge clk)
      if(!reset)
          state <= 4'b1000;
      else
          if(counter == 0)
              if( state<= 7 && state>= 0)
                  state <= state - 1;
              else
                  state <= 4'b0111;

  always @(state)
      case(state)
      4'b0000 : led = 8'b1111_1110;
      4'b0001 : led = 8'b1111_1101;
      4'b0010 : led = 8'b1111_1011;
      4'b0011 : led = 8'b1111_0111;
```

```
        4'b0100 : led = 8'b1110_1111;
        4'b0101 : led = 8'b1101_1111;
        4'b0110 : led = 8'b1011_1111;
        4'b0111 : led = 8'b0111_1111;
        default : led = 8'b1111_1111;
        endcase

  endmodule
```

图 48 led_core.v 源码

```
`timescale 1ns / 1ps
module test();
  reg clk,reset;
  wire [7:0] led;
  led_core led_core( clk, reset ,led );
  initial begin
        clk = 0;
        reset = 0;
        #1000 reset = 1;
  end
  always #1 clk=~clk;
endmodule
```

图 49 test.v 源码

3） 编译

点击【Compile】->【Compile All】对以上两个文件进行编译，当编译通过时在文件右边的 status 栏会打钩，如图 50 所示。如果编译不通过则需要对程序进行修改直到通过才能进行下一步。

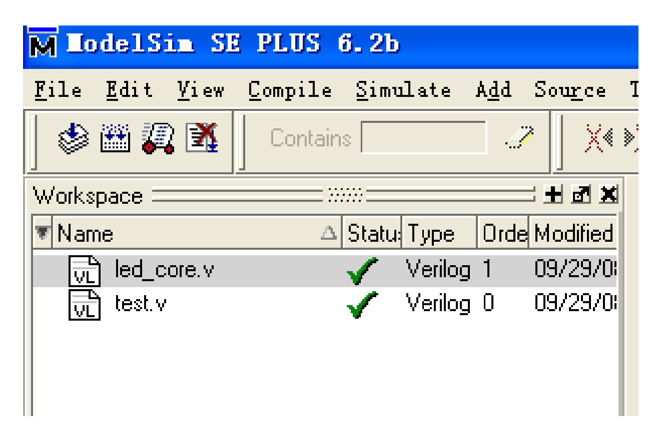


图 50 文件编译

4） 仿真

点击【Simulate】->【Start Simulate】，会出现 Start simulation 对话框。如图 51 所示，在 work 目录下选择顶层文件 test，点击 OK。

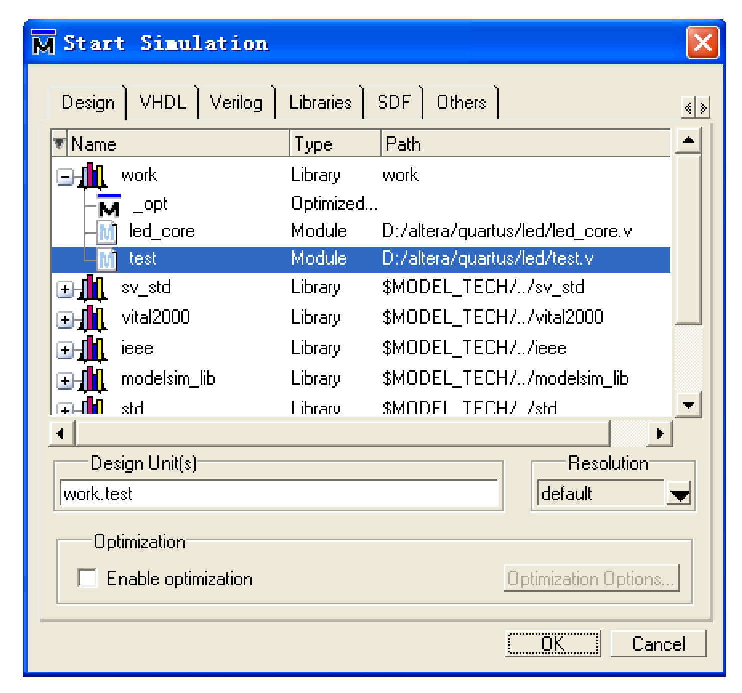


图 51　测试文件选择

这时在 workspace 下出现 instance 框，如图 52 所示。4 在 test 上点击鼠标右键，在 Add 选项中选择 Add to Wave。这样在仿真结束后会显示出仿真波形。

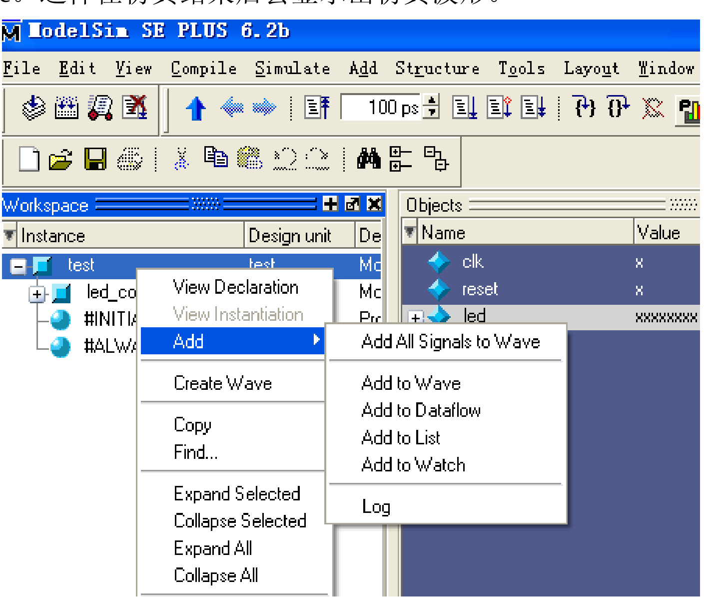


图 52　波形添加

这时在【Simulate】->【run –all】选项，开始仿真，而此菜单下的 Break 选项可以使仿真停止。仿真结束后，在 Wave 窗口里会出现仿真的波形，如图 53 所示。

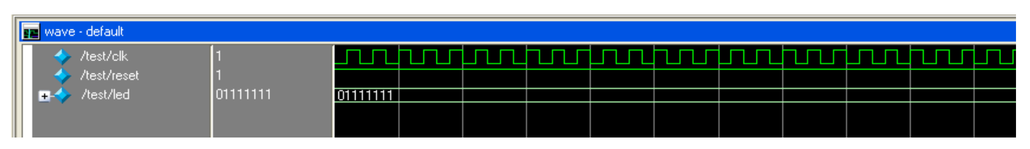


图 53　仿真波形

4．波形信号的保存

Wave 窗口内的仿真波形，可以保存起来以后调入。选择 File 菜单下的 Save format，保存成 wave.do 文件。如图 54 所示，需要调入时，在 modelsim 命令行内执行命令 do wave.do 即可。

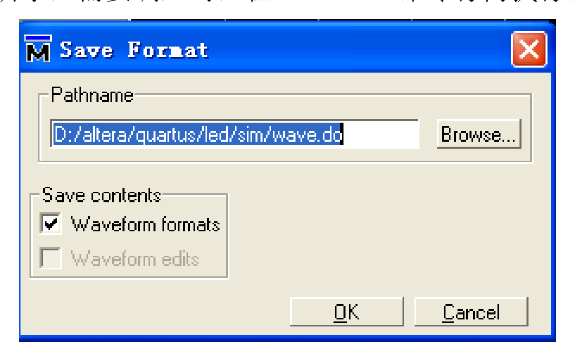


图 54　波形文件保存

6.3.2　加入 Altera 器件的仿真库

1） 找到 modelsim 的安装目录下的 modelsim.ini 文件，取消它的只读属性。

2） 打开 modelsim，在主窗口选择【File】→【Change directory】，将工作目录改变到你想存放仿真库的目录，点击 OK。这里选择 modelsim 的安装目录，如图 55 所示。

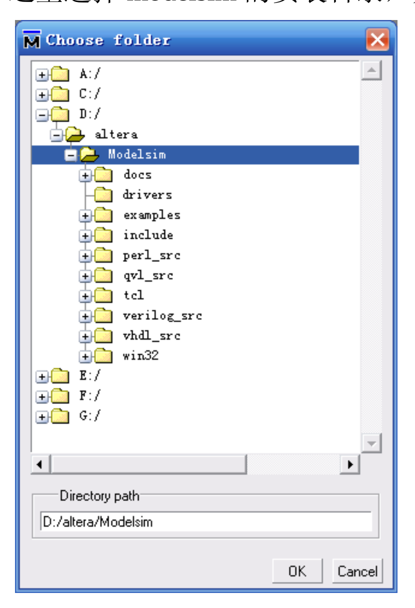


图 55　路径切换

3） 在生窗口中选择【File】→【New】→【Library】命令,在弹出的【Create a new library】窗口中将选项【Create】设置为【a new library and a logical mapping to it】,在【Library name】和【Library physical name】中键入所要创建库名字 altera_lib,如图 56 所示,此时在主窗口中已多了一个 altera_lib 项。

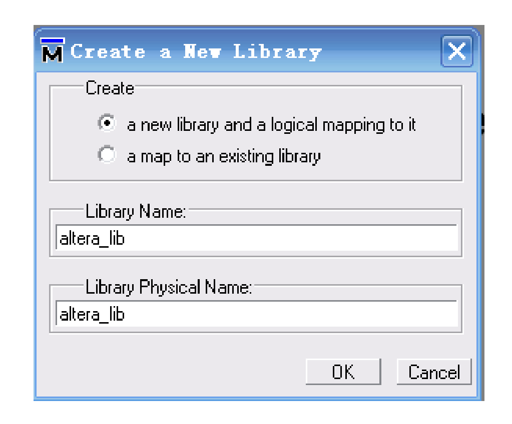


图 56　创建新库

4） 在 Workspace 中的 Library 中选中你创建的库 altera_lib,在主菜单中选【Compile】→【Compile…】;在弹出窗口 compile source file 窗口中的【Libaray】下拉菜单中选中你的库名,在【查找范围】中选择 quarturs 安装目录\quarturs\eda\sim_lib 文件夹下,对它前 8 个文件进行编译,注意需要编译两次，具体办法是先选择这 8 个文件,点击【Compile】,这次会出错,完成后再点击【Compile】，编译成功，点击【Done】，如图 57 所示。

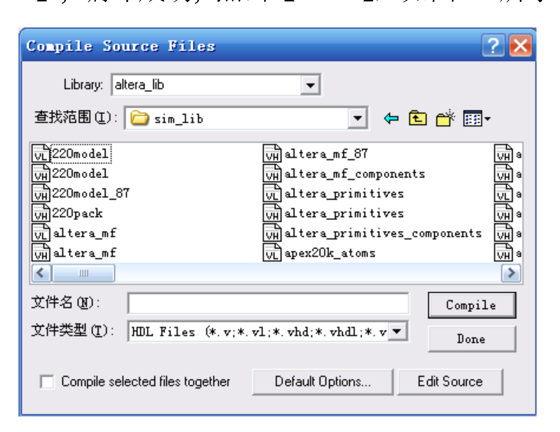


图 57　库的编译

5） 关闭 Modelsim，然后重新启动 Modelsim，如果这时 Workspace 中出现“altera_lib(unavailable)”，这时需要指定 altera_lib 的路径，在 altera_lib 图标上点击右键选择【edit】指定 altera_lib 库的路径，如图 58 所示。

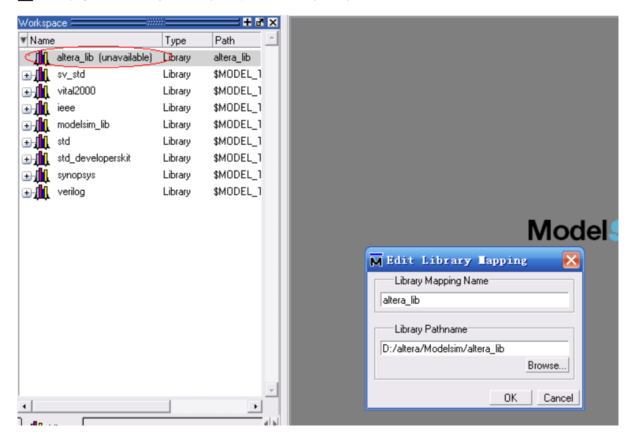


图 58　指定库的路径

6） 最后将 modelsim.ini 的属性重新改为只读。这样每次使用 Modelsim 仿真时就不用再对 Altera 库进行编译了。

7 常见问题解答

7.1 为什么我无法以 JTAG 方式下载程序？

1） 确认电源已经打开；
2） 确认 USB 线已经正确连接；
3） 确认模式开关（J15）处在 RUN 状态。

7.2 为什么我无法以 AS 方式下载程序？

1） 确认电源已经打开；
2） 确认 USB 线已经正确连接；
3） 确认模式开关（J15）处在 PRG 状态。

7.3 为什么串口无法打印信息？

1） 确认串口线正确连接，注意黑线是地线；
2） 确认超级终端参数是否正确；
3） 确认超级终端已处于连接状态。

8 演示程序的使用

8.1 核心板演示程序

1．LED 测试程序
   a) 演示程序：DE4.2_coremodule_ledtest_OK。
   b) 演示步骤：直接下载即可。
   c) 演示效果：上面板的 LED 有规律地闪烁。
2．uCLinux 测试程序
   a) 演示程序：
   b) 演示步骤：
      i. 如果是新板子，操作系统内核和文件系统已经烧进 Flash 中；如果 Flash 被写过，参考实验指示书重新下载内核和文件系统。
      ii. 将串口连接至计算机。
      iii. 打开超级终端，按以下参数设置：

波特率：　　115200 波特/秒
数据位：　　8 位
停止位：　　1 位
奇偶校验：　无
流控：　　　无

iv. 如果需要使用以太网，将网线接至网口；计算机 IP 设为 192.168.1.237（核心板上 IP 已缺省设为 192.168.1.236）。

v. 打开核心板的电源并下载 NIOS。

c) 演示效果

从超级终端上可以看到 uCLinux 正常启动，用户可以登录执行各种命令。

账户为：root；密码：uClinux

可以 ping 192.168.1.236 来确认网络是否正常。

8.2 扩展板演示程序

接通扩展板电源，按下扩展板电源开关，扩展板进入自检演示程序（必要时按动核心板复位按键 RESET）。

演示内容包括：

8.2.1 核心板、扩展板 LED 灯演示二进制加法运算；拨动扩展板开关 SW1、SW2 二进制运算速度将有所变化。

8.2.2 按动扩展板 4X4 开关，扩展板 1、2 位 7 段数码管显示按动的开关行列位置

8.2.3 扩展板 3、4 位 7 段数码管连续显示计数器数值

8.2.4 扩展板 7 段数码管 5~8 位显示 TFT 彩色显示屏上电阻式触摸屏 16 进制坐标，触动触摸屏坐标值将有所变化

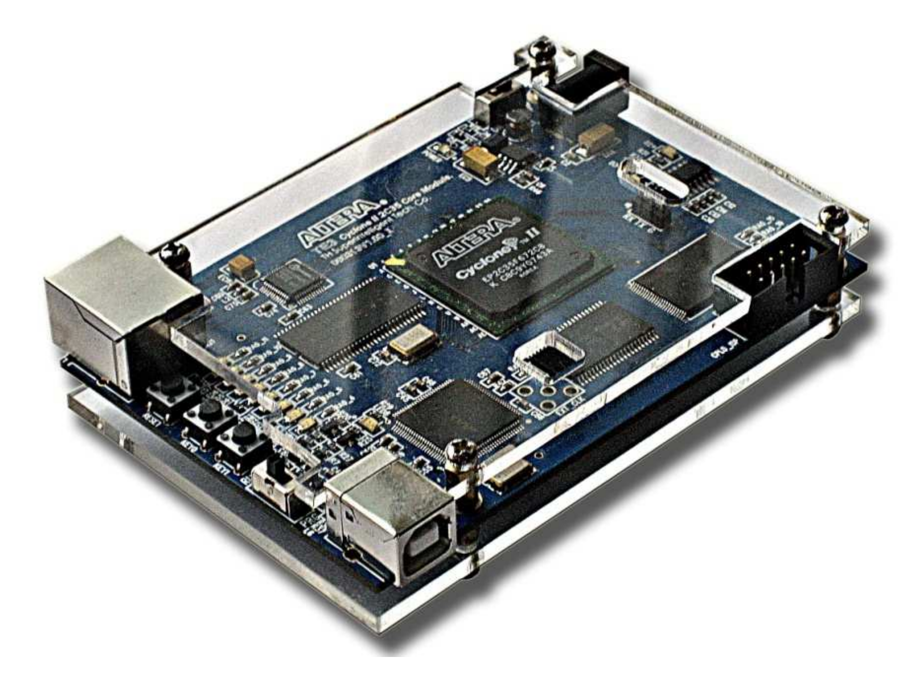
8.2.5 将音频信号接入扩展板蓝色音频输入插口，将耳机或者音频播放设备接入绿色音频输出接口将会收听到相应的音频信号

9 附件

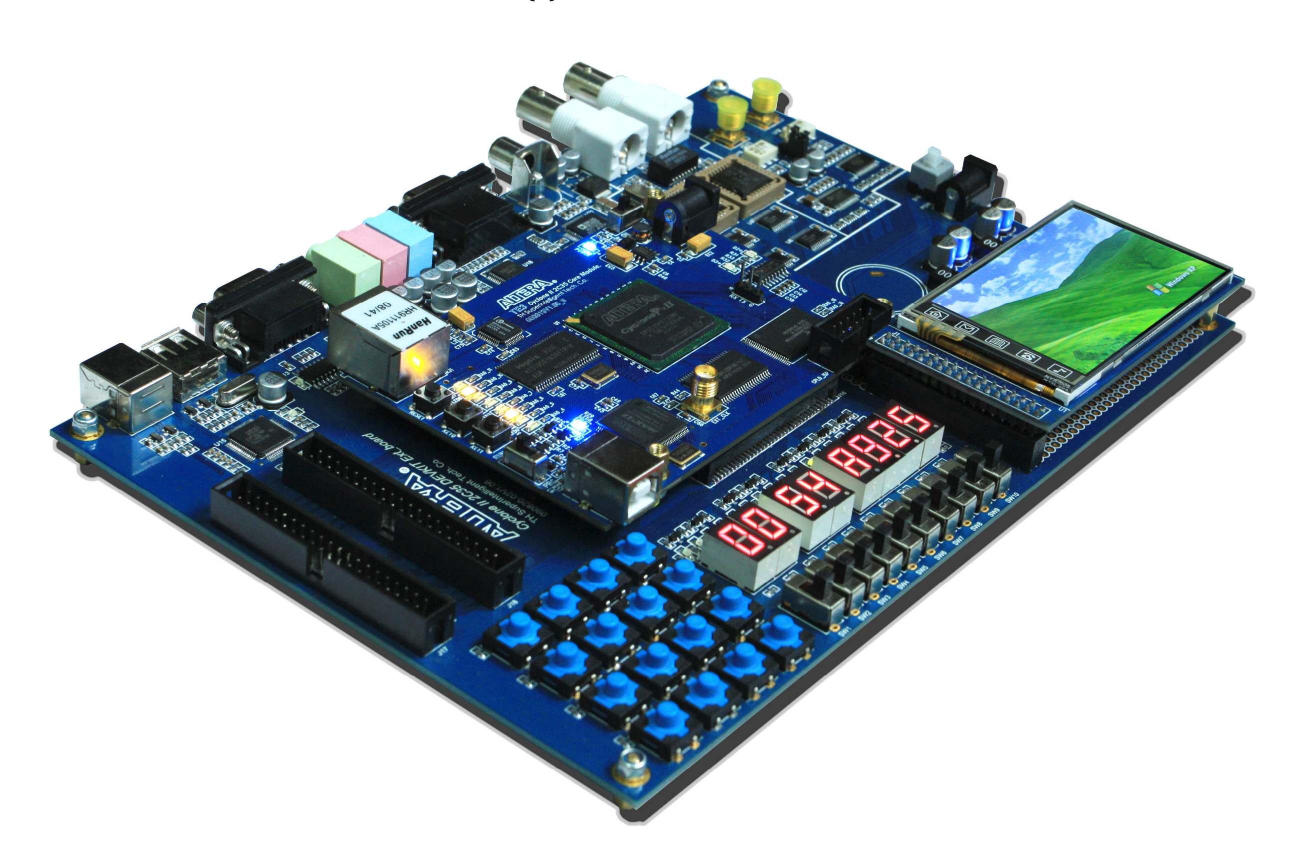
整套系统附件如所示。

表 26　附件列表

名称	数量
外包装盒	1 个
核心板	1 块（图 59(a)）
扩展板（可选件）	1 块（图 59(b)）
电源	1 个（图 59(c)）
下载线	1 条（图 59(d)）
网线（可选件）	1 条 (图 59(d)）
三针串口线(可选件)	1 条（图 59(f)）
说明书	1 本
说明光盘	1 张



(a) 核心版



(b) 核心板+扩展板



(c) 电源



(d) 下载线



(e) 网络线



(f) 串口线

图 59　套装附件

10 附录

[1] 管脚映射表

[2] 核心板原理图

[3] 扩展板原理图

[4] 实验指示书

[5] uClinux-v1.4.1，用在 NIOS II 上的 uClinux

以上信息可能会随着硬件、软件版本升级会有所变动，请随时访问相关网站：http://www.tseda.com/